Soft Detection of Trellis Coded CPM in Frequency-Selective Channels

Tri Pham

Electrical and Computer Engineering Department

University of Canterbury

A thesis submitted for the degree of

Master of Engineering (ME)

March 2012

Abstract

Non-linear continuous phase modulation has constant envelope and spectral efficiency, which are desirable for public safety communication systems where both bandwidth and power are limited. A practical design of an innovation based receiver for partial response CPM was recently developed for public safety applications. It is in the form of a linear predictive demodulator with a coefficient look up table. The demodulator shows great performance over multipath fading channels without channel equalization and promises a significant contribution to public safety communication.

The work in this thesis is focussed on developing and analyzing modern techniques to improve the receiver performance while maintaining a feasible implementation complexity. Suitable soft output algorithms are incorporated into the demodulator allowing a subsequent convolutional decoder to perform soft decoding. By modifying the design criteria of the predictive demodulator and introducing a feedback loop, an iterative detection scheme is formed for the concatenated structure of demodulator, deinterleaver and decoder.

Spatial diversity combining techniques are summarized and a very low complexity combining scheme is developed. It selects the best received sample sequence by considering the average energy of each sequence. In addition, the demodulator is extended to have dual coefficient look up tables supporting its detection by having parallel prediction processes and combining their results. This leads to an improvement in overall demodulator performance. A theoretical proof that only half the number of coefficients need to be stored in memory is also given.

Matlab simulations on a Rayleigh fast fading multipath channel have shown that the proposed techniques significantly improve the overall detection accuracy. Each of them provides a good gain in signal to noise ratio or delay spread and when combined, a significant performance gain is achieved.

This thesis is dedicated to my family, whose love and support have given me strength to go along the research journey.

Acknowledgements

It is an honor for me to acknowledge the help of many people, without which this thesis would not have been possible.

First, I would like to show my deep gratitude to my principal supervisor, Dr. Philippa Martin, whose most outstanding help in my research was to provide me with great insight into the area of error control coding. In addition, her caring attitude and her constant encouragement and inspiration helped to keep me on task through out.

I am indebted to my second supervisor, Professor Desmond Taylor, whose admirable knowledge and teaching steered me towards the area of wireless communication during my undergraduate years. For this thesis, Professor Taylor made his support always available in various ways, and he gave me lots of invaluable suggestions and research materials in the area of signal demodulation.

I am fortunate to have had Dr. Clive Horn as my third supervisor. His tireless assistance, especially with software development and debugging, and the many constructive discussions on practical radio systems I had with him were definitely indispensable in my research.

I would never forget Professor Jim Cavers for his valuable input in setting up the thesis research project and his enlightening advice in the project progression.

I am especially thankful for the generous funding I received from the Foundation for Research, Science and Technology for this project. I am also grateful to the Wireless Research Center and Tait Electronics Ltd. (particularly the terminal DSP team), who provided me with sufficient resources and assistance throughout the course of this research.

I appreciate my many supportive friends who offered me useful ideas for this project and thesis writing. Alice Chu, Michael Krause, Rui Lin, Tauseef Khawaja and Zahid Rauf in the Communications Laboratory at Canterbury University deserve special mention.

Most importantly, I wish to thank my family, including my little sister, Hannah Jones, for providing a loving environment for me. I acknowledge my stepfather, Greg Jones, for his frequent support, especially helping me with my English writing. I owe my mother, Lan Jones, for her endless care and love, which have brought me to where I am in my life journey.

Lastly, my thanks to all those whose names I am unable to mention here but supported me in many respects during the completion of the thesis.

Contents

List of Figures

List of Tables

Abbreviations and Acronyms

APCO Association of Public-Safety Communications Officials.

AWGN Additive White Gaussian Noise.

BER Bit-Error Rate.

BLER Block-Error Rate.

C4FM Continuous 4 Level Frequency Modulation.

CPM Continuous Phase Modulation.

DMR Digital Mobile Radio.

DPSK Differential Phase Shift Keying.

ECE Error Control Encoding.

EGC Equal Gain Combining.

FEF Front End Filter.

FIR Finite Impulse Response.

GMSK Gaussian Minimum Shift Keying.

GSM Global System for Mobile communications.

HCPM Continuous Phase Modulation.

ISI Inter Symbol Interference.

LOVA List Output Viterbi Algorithm.

LUT Look Up Table.

MAP Maximum A Posteriori.

MIMO Multiple Input Multiple Output.

MLSE Maximum Likelihood Sequence Estimator.

MMSE Minimum Mean Squared Error.

MRC Maximum Ratio Combining.

MSK Minimum Shift Keying.

P25 Project 25.

PEF Prediction Error Filter.

PSD Power Spectral Density.

PSP Per-Survivor Processing.

RC Raised Cosine.

RLS Recursive Least-Squares.

RSSE Reduced State Sequence Estimation.

RSSI Received Signal Strength Indication.

SC Selection Combining.

SNR Signal to Noise Ratio.

SOVA Soft Output Viterbi Algorithm.

SSC Sequence Selection Combining.

SSPE Sum of Squared Prediction Errors.

VA Viterbi Algorithm.

Chapter 1

Introduction

1.1 General Overview

Today's information society has a heavy demand for communication systems that are cheaper, faster and of higher quality. The proliferation of wireless communication services for voice, video and data, has necessitated the development of systems that have greater capacity and can operate reliably over the often hostile wireless channel. In order to achieve greater capacity, we can employ larger channel bandwidth, larger transmit power and/or more spectrally efficient methods for transferring information. Commercial mobile phone systems use packet-switched and large bandwidth technologies to support a broad range of multimedia applications [1]. In contrast, due to limited spectrum and regulatory restrictions, the latter is often the more viable and cost effective option for public safety land mobile radio systems [2, 3].

Continuous phase modulation (CPM) is a constant envelope modulation technique in which memory is added in the phase modulation process to ensure that the phase is a continuous function of time [4, 5, 6]. CPM possesses many desirable features for mobile communication systems where both bandwidth and power are limited. Since the modulation has a constant envelope, it can use high efficiency and low cost non-linear power amplifiers instead of linear amplifiers.

The phase memory in CPM signals allows the modulation to have better spectral characteristics and better transmission reliability than other types of constant envelope modulation that have no memory such as M-ary Phase Shift Keying. Hav-

ing these attractive features, CPM has been deployed in many mobile systems such as *global system for mobile communication (GSM)* [7], *digital mobile radio (DMR)* [8], and *association of public-safety communications officials (APCO)* systems [2].

A CPM signal can be either coherently or differentially detected. In coherent detection, the phase state of the signal is included in the detection process, while it is left out in most differential detection schemes. In both types of CPM detection, the sequence of the signal phases at different symbol times forms a path through its corresponding phase trellis. Hence, the *Viterbi Algorithm (VA)* can be applied to perform *maximum likelihood sequence estimation (MLSE)* of the data in the received signal [6, 9].

In practical CPM systems, convolutional codes are often applied to reduce the *signal to noise ratio (SNR)* required for reliable data detection [10]. Hence, a classical CPM receiver usually contains a demodulator that is followed by a convolutional decoder [11, 12]. Depending on the type of input to the decoder, it can apply hard or soft decision decoding of the data signal. Between the two methods, soft decoding is favored as it produces better estimates of the transmitted data.

A new demodulation scheme targeted for public safety multicast applications was recently developed for single modulation index CPM (which is often referred to as *single-h* CPM where h is the notation for the modulation index) such as *continuous 4 level frequency modulation (C4FM)* used in APCO-Phase 1 and *Harmonized CPM (HCPM)* used in APCO-Phase 2 [2]. The new design is different from the traditional approaches in the sense that the receiver actively computes linear prediction of the incoming received sample based on the previous received samples. The prediction can be done because the memory of CPM produces dependence between different samples in the transmitted signal. The wireless channel disturbs this dependence, and the linear predictions then become an approximation to the mean of the actual received sample. The difference between the predicted sample and the actual one can be used as a reliable metric for MLSE demodulating the received modulated signal [13, 14]. Thus, this design is able to outperform traditional coherent and differential demodulation receivers, especially in frequency selective channels in which the delay spread can dramatically reduce signal detection accuracy [14].

Although, the original design has shown very good performance, the capability limit of this receiver structure has not been fully exploited. For example, the predictive demodulator processes its CPM trellis with a VA, whose hard bit decisions are fed through a deinterleaver to the VA of a subsequent convolutional decoder. As a result, the decoder has to apply a hard decision decoding scheme based on Hamming distance, which is distinctly sub-optimal [15].

1.2 Research Direction

The core of this project is to develop and evaluate suitable methods for enhancing the predictive demodulator and the overall reception system. From the previous discussion, it is clear that one of the most applicable improvements here is soft detection, which involves the generation of soft information from the demodulator. An important part of this research is the development of low and high complexity algorithms, for the predictive demodulator to produce soft output information, which can be used to strengthen the *block error rate (BLER)* and *bit error rate (BER)* performances of the following convolutional decoder.

The low complexity algorithm should be able to provide reasonable gain in the reliability of signal detection; and also needs to be simple so that it (and possibly an iterative structure based on this design) can be implemented on mobile communication devices in which computational power is limited. On the other hand, the high complexity one is aimed at achieving the best performance for the predictive demodulator via the exploitation of reliability information. This provides a good reference for performance comparison with the lower complexity algorithm. Hence, computational complexity is not a restriction for the design of the second algorithm. The thesis will look at how the developed algorithms perform under various mobile channel conditions as well as their computational expense. Finally, the knowledge gained will be used to investigate iterative receiver processing techniques.

Another suitable approach for increasing the system accuracy is to apply spatial diversity at the receiver. One method of employing receiver diversity has already been explored in [13]. This method is, however, quite complex as the computational load of the demodulation process increases linearly with diversity order. Here, we present a much simpler diversity approach, which involves the selection of the best

received sample sequence before the demodulation process takes place. This technique allows the receiver to utilizes diversity without adding a significant computation load to the reception process. The final method considered here is to have multiple parallel prediction processes in the demodulator structure. The outcome of these independent predictions can be jointly used to achieve a better demodulation process.

1.3 Literature Review

1.3.1 Reliability Generation Algorithm

Generating soft information from an operation that involves a trellis is not a new topic of interest. Given that the trellis constructed for the predictive demodulator is non-binary and is in the form of a reduced-state trellis (as discussed in Chapter 4) similar to those in [16]-[18], there are many algorithms available in the literature for performing such a task; these can be classified into sub-optimal (often with low complexity) and optimal (usually with high complexity) schemes.

The two most famous sub-optimal approaches are perhaps the *list output Viterbi algorithm (LOVA)* and the *soft output Viterbi algorithm (SOVA)*. The LOVA algorithms [19]-[27] are generalizations of the VA in order to compute an ordered list of the S best paths ($S = 1$ corresponding to the VA). In LOVA, the symbol decisions along each path are hard, but the probability of each symbol sequence is provided. The S symbol sequences are ordered according to their probabilities in the list. These soft sequence decoding algorithms can provide significant performance gain over the VA in many different concatenated structures such as turbo decoding [26, 27]. However, as shown in [24], LOVA complexity increases at least linearly with S and good performance is usually obtained only when S is reasonably high. These features together with a potentially high number of states in the non-binary demodulation trellis can make the soft information generating process very computationally demanding. Hence, LOVA algorithms might not be suitable candidates for the sub-optimal algorithm.

The concept of SOVA is quite different to that of LOVA. Instead of considering a number of sequences, SOVA only considers the maximum likelihood sequence and

calculates the likelihood value of each symbol in that sequence [24]. The original SOVA algorithm was presented by Hagenauer and Hoeher in [28]. This algorithm was developed for a binary trellis in which there are only two paths competing for survival at each trellis state. The simple extension to a non-binary trellis was proposed in [29, 30]. In this extension, at every state on the survivor path, the SOVA is applied to all the pairs of survivor paths and non-survivor ones. The complexity of such a process can be quite high if the number of paths entering each trellis state is large.

When comparing the performance of the binary SOVA to that of the optimal MAP algorithm, the performance degradation is small. However, its extension to a non-binary trellis is known to perform badly as shown in [31]. Better non-binary variations of the SOVA are shown in [31]-[34]. In [32], the proposed algorithm performance is near that of the optimal *maximum a posteriori (MAP)* algorithm. The SOVA variations in [33, 34] performed equivalently to the max-log MAP algorithm. The common characteristics of these algorithms are that both their memory and computational requirement increase linearly with the trellis size and the trace back length. This happens because they take into account all the paths entering each survivor state. Due to such complexity, these algorithms are less attractive for a serial concatenated structure such as considered in this thesis.

Another SOVA algorithm for non-binary trellises is described in [35]. This work is based on [33], but it introduces the process of converting symbol reliability into bit reliability. In [29] and [36], Turner develops a SOVA algorithm that delivers soft information by considering only the two best paths at each survivor state. Although Turner's algorithm cannot be directly applied to this application because of the difference between Turner's system and the one here, its concept is very attractive to our goal of designing a low-complexity sub-optimal algorithm. This is because such a concept can potentially eliminate the limitation of high computational requirement when the the trellis size is large.

In terms of an optimal algorithm, the most common and widely accepted is the symbol-by-symbol MAP algorithm (or BCJR algorithm) [37]. In [38]-[43], the BCJR is applied to reduced-state trellises. The BCJR algorithm is a good candidate for the proposed high complexity algorithm. This is because the BCJR algorithm has

been shown to provide outstanding performance in many applications. In addition, its derivation is very different to that of LOVA and SOVA, thus allowing the demodulator to explore its soft detection capability with two completely different soft information generation techniques.

1.3.2 Diversity Combining for CPM

Diversity combining is a technique that is used to combat multipath fading in a wireless channel. Receiver spatial diversity is realized when more than one antenna is used to receive the faded signals. The resulting multiple faded signals can be jointly used to achieve a better demodulation process. There are many different combining techniques available in literature. However, many of them are modified from the three classic combiners, which include *selection combining (SC)*, *equal gain combining (EGC)*, and *maximal ratio combining (MRC)* [44]. A more detailed description of these combiners can be found in the next chapter. Here, we provide a summary of the available literature on these methods applied for CPM systems.

The application of spatial diversity to CPM modulations with various demodulation detection schemes and channel models has been reported by a number of authors [47]-[54]. Some of the most relevant schemes are listed in Table 1.1. In general, the combining techniques can be ranked from best to worst performance as MRC > EGC > SC. On the other hand, their corresponding implementation complexities follow an inverted order of MRC < EGC < SC. MRC is the most complex scheme because it requires the estimation of the channel fading gain as well as noise variance at each antenna, while it is only required to approximate the channel phase offset for the less complex EGC and identify the strongest received signal in the case of SC [45, 46].

From Table 1.1, it can be seen that the use of diversity with variations of MSK and GMSK with differential and coherent detections have been commonly reported in the literature. However, currently no reports can be found on the application of diversity combining for a single-h CPM predictive system for fast fading multipath channels. In this Masters research, the combining techniques will be investigated to find the best practical combining technique for this application.

Table 1.1: Summary of the work done in the literature for CPM with diversity.

Reference	CPM Type	Demodulation Type	Combining Method	Channel Model
[47]	GMSK	Coherent	SC, EGC	Frequency selective
[48]	MSK	Differential	SC	2 paths Rayleigh
[49]	GMSK	Differential	MRC	Slow Nakagami
[50]	GMSK	Discriminator	SC	Nakagami
[51]	Partial response	Coherent	MRC	Rayleigh
[52]	MSK	Differential	SC, EGC, MRC	Rayleigh
[53]	CPFSK	Differential	MRC	Rician
[54]	Partial response	Coherent	SC, EGC, MRC	Fast Rayleigh

1.4 Scope

In this thesis, a number of techniques for improving the CPM predictive demodulator in [13] are developed. Soft ouput demodulators based on the SOVA and BCJR algorithm are developed as well as an iterative structure for closed-loop detection. The SOVA-based algorithm design is an extension of the ideas in [28, 36] to a general trellis operation. The BCJR-based algorithm is a modification of the algorithms in [37, 41, 43]; it applies the basic derivation steps of these algorithms, but takes into account the design characteristics of this particular demodulator. The iterative structure allows the predictive demodulator to exchange soft information with the trellis of an error control decoder. Here, for keeping the complexity as small as possible, the research only evaluates the performance of the iterative system implemented with the simple SOVA-based algorithm.

The other improvements include the application of a suitable diversity combining technique and the use of independently parallel predictions. In terms of diversity combining techniques, this research focuses on SC with 2 receive antennas. It is assumed that these antennas are spaced far apart so that their corresponding received signals are independent. As will be shown in Chapter 3, the prediction process relies on a group of coefficient sets, which is pre-computed based on certain assumptions about the transmission channel. By using each coefficient group for a particular parallel prediction and combining the results of these prediction processes at each time step, the demodulator can achieve better overall performance.

The examples and performance results presented in this thesis are focused on the P25 Phase 1 system whose modulation is C4FM [2]. The system primarily considers the rate 1/2 code used in the P25 Phase 1 standard as well as its interleaving scheme. A Rayleigh channel model with *additive white Gaussian noise (AWGN)* is assumed. The proposed schemes are for mobile and for long range communication. Therefore, it is assumed that the Rayleigh model is fast fading and has two paths in which the delay between them can be up to half the symbol period. Additionally, the receiver is assumed to have perfect knowledge of symbol timing, carrier frequency and carrier phase as the estimation of such synchronization parameters are beyond the scope of this thesis. The developments will be carried out using computer simulations, as opposed to the actual hardware design and implementation. In some sense, this study can be seen as a proof of concept or even a specification for an actual hardware implementation of the system.

The new predictive demodulator design and its enhancements may be implementable in real world wireless communication systems at an affordable cost and has considerable potential for commercial development in the area of rapidly deployable wireless data networks. The scheme could be developed to operate with other public safety digital radio or general mobile networks that utilize the CPM modulation or its variations.

1.5 Thesis Outline

Chapter 2 begins by introducing the general baseband communication link for mobile transmission. Continuous phase modulation is then reviewed with a specification of C4FM modulation. This is followed by a description of the wireless channel and its different models. In particular, attention is paid to the Rayleigh multipath fading channel, which can be represented by a discrete multipath model as shown in [62]. This model is based on the assumption that the delay power profile and the Doppler spectrum of the channel are separable. In other words, the complex path gains are uncorrelated with each other. The rest of the chapter is dedicated to discussions of the rate 1/2 convolutional code, inverleaver, and noise-limiting filter considered in this research. The final section explains diversity combining and its performance advantages over reception with a single antenna.

Chapter 3 contains a detailed summary of the predictive demodulator design. This includes the derivation of the detection process from first principles and the construction of the demodulation trellis as well as the complexity reduction method. The demodulation extensions to multiple samples per symbol and spacial diversity are also covered here. The final section shows the computation of the predictive coefficients and the possibility (as well as motivation) to have parallel prediction processes with the use of more than one coefficient *look up table (LUT)*.

Two soft output algorithms and an iterative structure for the demodulator are derived in Chapter 4. The derivations for these algorithms are based on the classic SOVA and BCJR algorithms [28, 37]. The developed algorithms share a commonality in which they both estimate symbol reliability before moving on to convert this into soft bits that can be processed by the subsequent decoder. The designation of the closed-loop detection involves a modification of the demodulator design so that it can take into account the feedback information, and a construction of a feedback loop for delivering suitable soft information (in the form of symbol probabilities) to the demodulator. Closed-loop detection performance is evaluated by applying the SOVA-based algorithm to both the demodulator and the decoder of the rate 1/2 code.

In Chapter 5, an investigation of a suitable diversity combining technique is carried out. The new diversity scheme performance is compared with the original diversity technique in both iterative and non-iterative detection. Following this is a proof that the receiver only needs to store half the number of coefficients required for its operation due to their symmetry. The chapter continues with the development of methods for utilizing multiple parallel predictions (or multiple coefficient LUTs) within the demodulation process.

Finally, the pertinent results of the thesis are summarized and some ideas for future research related to the work completed in this thesis are presented in the later Chapter 6.

1.6 Research Contributions

The original work in this thesis is formed by Chapters 4 and 5, and includes:

- A detailed investigation of the predictive receiver performance when used with a trellis code under fast fading multipath channel conditions.
- Developments of a low complexity SOVA-based algorithm that can deliver good performance across a wide range of channel conditions and a high complexity BCJR-based algorithm that shows outstanding performance with low delay spread for C4FM or equivalent signal modulation.
- Development of a general iterative detection scheme for the serial concatenation of the predictive demodulator and convolutional decoder. Excellent performance is obtained when implemented with the SOVA-based algorithm.
- Designing a SC method that is significantly simpler than the original diversity technique while its performance is close to that of the original one when incorporated with iterative detection.
- A proof that only half the number of coefficients are required. In addition, a parallel prediction scheme for the demodulation process is developed by using dual LUTs at the same time. Considerable gains in SNR or delay spread can be obtained with this method.

Chapter 2

General Background

In this chapter, the relevant background information for the CPM system considered in this thesis is presented. First, a description of a general communication link, which includes all the signal processing parts between the input data at the start of transmission and the output message at the end of reception, is given. The following section introduces CPM modulation and its demodulation; particular attention is payed to C4FM, which is one of the main modulations used in public safety communication standard APCO-Phase 1 [2] (the CPM predictive demodulator design is described later in Chapter 3). It is followed by a discussion of the wireless multipath channel and the fading environment. A multipath fading channel can have devastating effects on the performance of communication systems and it is important to know its characteristics. In the three subsequent sections, the convolutional codes, interleaver, and noise-limiting filter considered in this research are defined. The final section here provides an overview about different diversity methods to combat multipath fading and hence improve system performance.

2.1 Communication System Overview

Figure 2.1 shows a schematic of a general communication link. The purpose of a communication system is to deliver a message from an information source in recognizable form to a user destination. A communication system is divided into three essential parts, namely, the transmitter, the receiver, and the medium through which the data is transferred, known as the channel, where noise and distortion can take place. In designing any communication system, it is important to keep the trans-

mission reliability as high as possible. In order to achieve this, various methods have been employed in system construction. In terms of baseband signal processing, the core techniques for data protection includes modulation, error control coding, and interleaving. A noise-limiting filter is included at the receiver front end to partially reduce the effect of noise and interference in the received signal.

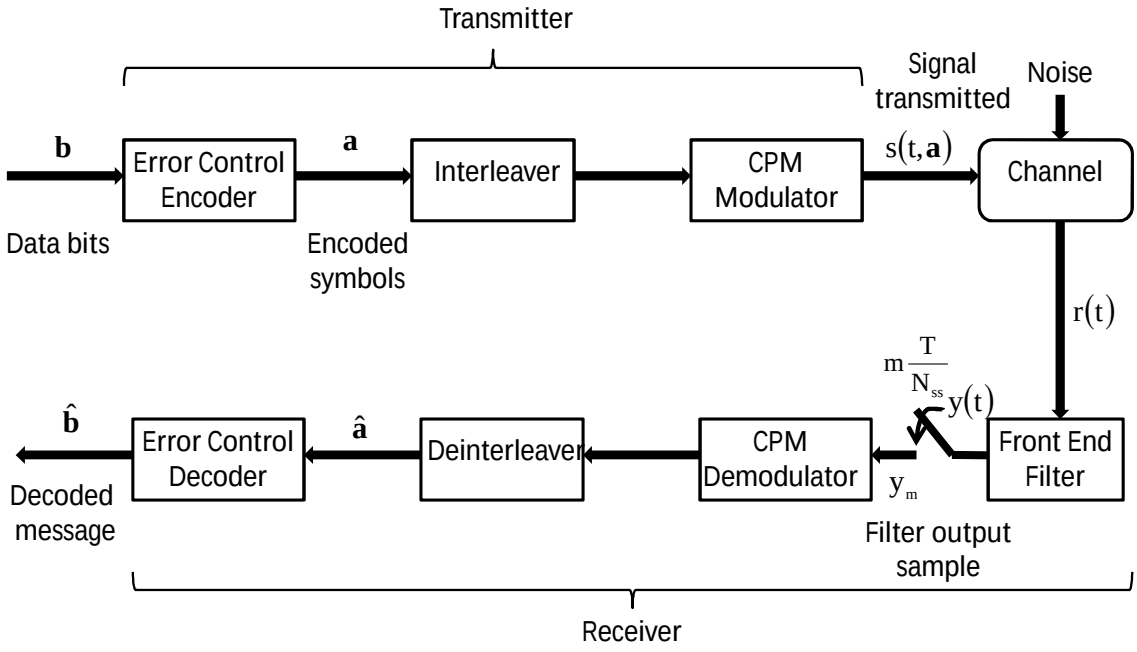


Figure 2.1: A general wireless communication system.

Modulation is the process of modifying the message signal into a form that is suitable for transmission over the channel. This modification is achieved by varying some parameters of a carrier wave in accordance with the message. In the case of CPM modulation, the carrier phase is used for data transmission. The receiver re-creates the original message signal from a degraded version of the transmitted signal after propagation through the channel. This re-creation process is known as demodulation, which is the reverse of the modulation process. However, because of the unavoidable presence of noise and distortion, it is common for the demodulation process to not be able to re-create the original signal message exactly. The resulting degradation is influenced by the type of modulation used as well as the demodulation method. In the case of CPM and many other modulations, MLSE and MAP trellis demodulators have proven to produce optimal performance [56].

Because of the often hostile radio channel, modulation alone can not guarantee a reliable re-creation of the original message (even with high performance demodulation methods such as MLSE or MAP). Additional protection schemes must be included in the system. *Error control encoding (ECE)* is the standard data protection technique in many communication systems. ECE is based on the idea that error detection and correction for the demodulated signal can be achieved by adding some redundancy to the original message. Such redundancy can be used by receivers to check consistency of the delivered message and to recover data determined to be erroneous. Convolutional codes are an effective form of ECE that transforms the original message into an encoded message that has at least as many bits as the original message [56]. The processes of convolutional encoding and decoding are both done on a trellis. As mentioned earlier, the performance of a convolutional decoder depends on the type of its input in which soft symbol/bit is superior to hard symbol/bit; hence, there is a need for the earlier process of demodulation to produce soft instead of hard decisions. This is a strong motivation behind this research.

Although a convolutional code is a great method to identify and correct errors in the demodulated data, its performance degrades significantly when these errors appear in bursts, which may often be the case. To help the decoding process, an interleaver and deinterleaver are included in the transmitter and receiver as shown in Fig. 2.1. These devices can effectively split the error burst apart, thus increasing the decoding accuracy.

2.2 Continuous Phase Modulation and C4FM

Continuous phase modulation, or CPM, is a non-linear, constant envelope modulation [4, 6]. CPM is well suited for transmission of digital signals over power and band-limited channels, such as those for mobile land, satellite, and radio channels. The phase continuity of CPM, which results from the modulation memory, improves its spectral efficiency by maintaining smooth phase transitions, rather than the abrupt phase changes exhibited by modulations without phase continuity.

CPM is a true constant envelope modulation, which allows the use of low cost,

power efficient, non-linear power amplifiers without introducing distortion. Conversely, if a modulation envelope varies with time and it is transmitted using a non-linear power amplifier, there will be non-linear distortion in the received signal. To reduce distortion for such modulations, expensive linearized amplifiers can be used, or the output power must be significantly reduced, which means that the amplifier can not operate at its peak efficiency [65].

The complex envelope of a single modulation index CPM signal $s(t, \boldsymbol{a})$ is given by

$$s(t, \boldsymbol{a}) = \sqrt{\frac{E_s}{T}} \exp\left(j\phi(t, \boldsymbol{a})\right), \tag{2.1}$$

where the signal phase $\phi(t, \boldsymbol{a})$ is computed as

$$\phi(t, \boldsymbol{a}) = 2\pi h \sum_{k=0}^{n} a_k q(t - kT), \tag{2.2}$$

and $nT \leq t \leq (n+1)T$, a_n is the n^{th} modulation symbol, $\boldsymbol{a}$ is the entire transmitted sequence at time nT, h is the modulation index, $q(t)$ is the phase-smoothing function, T is the symbol time, and E_s is the energy per transmitted symbol. The symbols a_n are assumed to be independent and identically distributed and from an A-ary set given by

$$\Omega_a = \Big\{ \pm 1, \pm 3, ..., \pm A - 1 \Big\}, \tag{2.3}$$

where A is usually a power of 2. The relationship between E_s and the energy per bit E_b is $E_s = E_b \log_2(A)$.

In order to achieve good spectral properties $q(t)$ has some special features. It is continuous, non-decreasing, and

$$q(t) = \begin{cases} 0 & , t \leq 0 \\ \\ \frac{1}{2} & , t \geq LT, \end{cases} \tag{2.4}$$

where L is the effective phase response length of the modulation. Since $q(t)$ has a smooth shape without discontinuity that affects the phase transition over L symbols, the resulting phase response $\phi(t, \boldsymbol{a})$ (which is the sum of different scaled versions of $q(t)$ over time according to 2.2) is also a smooth function. Therefore, there is no abrupt change in the CPM signal phase that can lead to high frequency components and the signal can fit in a narrow bandwidth. In order words, the CPM signal

achieves spectrum efficiency.

In practice, the phase filtering process can be constructed as a combination of frequency pulse shaping and integration. As the length of $q(t)$ is specified by L, the length of the frequency pulse $g(t)$ is also given by LT. The modulation index h determines both the bandwidth of the transmitted signal and the distance properties of the resulting signal set. Because $q(LT)$ is normalized to be 1/2, (2.2) can be written as (for $n \geq L$ and $nT \leq t \leq (n+1)T$)

$$\phi(t, \boldsymbol{a}) = \pi h \sum_{k=0}^{n-L} a_k q(t - kT) + 2\pi h \sum_{k=n-L+1}^{n} a_k q(t - kT) \tag{2.5}$$

$$= \theta_n + 2\pi h \sum_{k=n-L+1}^{n} a_k q(t - kT). \tag{2.6}$$

In (2.6), the phase state, θ_n, is common to all choices of L symbols a_{n-L+1} to a_n.

From the above formulation, we can see that the actual form of CPM is defined by the pulse length L, the modulation index h, the number of data levels A, and the phase pulse $q(t)$ (or the frequency function $g(t)$). In terms of L, the family of CPM can be divided into two classes: full response CPM and partial response CPM. Full response CPM is defined with $L = 1$; in other words, its pulse shaping function only covers one symbol interval. On the other hand, partial response has $L > 1$ that provides additional memory to the modulation scheme; hence, it has better spectral efficiency than that of the full response CPM without performance degradation [6]. The modulation index h defines the maximum (peak to peak) frequency deviation of a CPM signal, which determines the phase change rate [9]. CPM systems that use a fixed value of h for all transmissions are referred to as single-h CPM while other CPM systems, which utilize a set of different h values, are referred to as multi-h CPM. In general, multi-h CPM schemes provide better transmission reliability, but at the cost of increased complexity and a wider spectrum [9].

The parameter A of a CPM signal specifies the number of valid symbols in the modulation scheme and, hence, partially affects the system data rate. However, if the data rate is fixed, increasing A generally has two effects, namely, it decreases the main lobe spectral width, and increases the susceptibility to noise. In such cases, changing the value of A introduces a trade off between error performance

and spectral efficiency [6]. The frequency pulse $g(t)$ also plays an important role in the classification of CPM as it (with the help of an integrator) transforms the data symbol into signal phase such that the transmitted signal spectrum can be confined to the limited bandwidth available. Therefore, $g(t)$ must be a smooth function with no discontinuity or impulses, so that the resulting phase output is continuous. Some commonly used pulse shapes include rectangular, *raised cosine (RC)*, and Gaussian functions.

By changing the four parameters defining CPM, a wide variety of CPM modulation schemes can be devised. As the predictive demodulator is designed for emergency public safety communication, the present research considers C4FM, a particular type of CPM that has been deployed in APCO P25 Phase 1. C4FM is a 4-level CPM that has a single modulation index $h = 0.25$. The modulation utilizes a frequency pulse filter whose frequency response is a RC function given by

$$G(f) = \begin{cases} 1, & |f| \leq 1920Hz \\ \frac{1}{2} + \frac{1}{2}\cos\left(\frac{\pi f}{1920}\right), & 1920 < |f| \leq 2880Hz \\ 0, & |f| > 2880Hz. \end{cases} \tag{2.7}$$

For $L = 3$, the corresponding frequency and phase pulses in the time domain $g(t)$ and $q(t)$ are shown in Fig. 2.2.

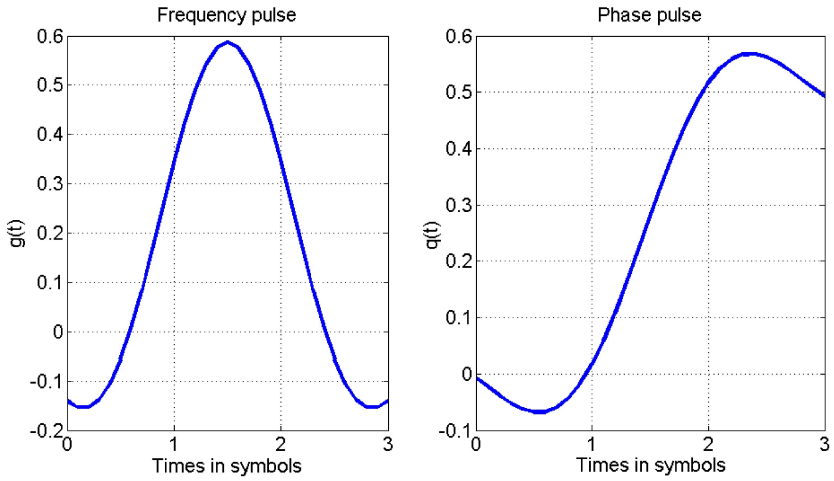


Figure 2.2: C4FM frequency and phase pulses.

2.3 Wireless Channel

The performance attainable by wireless communication systems is limited by the effects of the radio channel. The channel has a time-varying nature, and hence transmission and analysis is more difficult than for fixed wired channels [58]. As shown in Fig. 2.3, transmission paths may be obstructed by natural and man-made objects such as hills, buildings and trees. Such obstructions in the environment cause multiple reflections of the transmitted signal to be received from different directions and with different propagation delays. Interaction between the waves causes a phenomenon known as multipath fading, which can result in severe and rapidly fluctuating attenuation of the transmitted signal. This may result in an inability to reliably transfer information. In the following sub-sections, we discuss possible forms of the multipath fading channel.

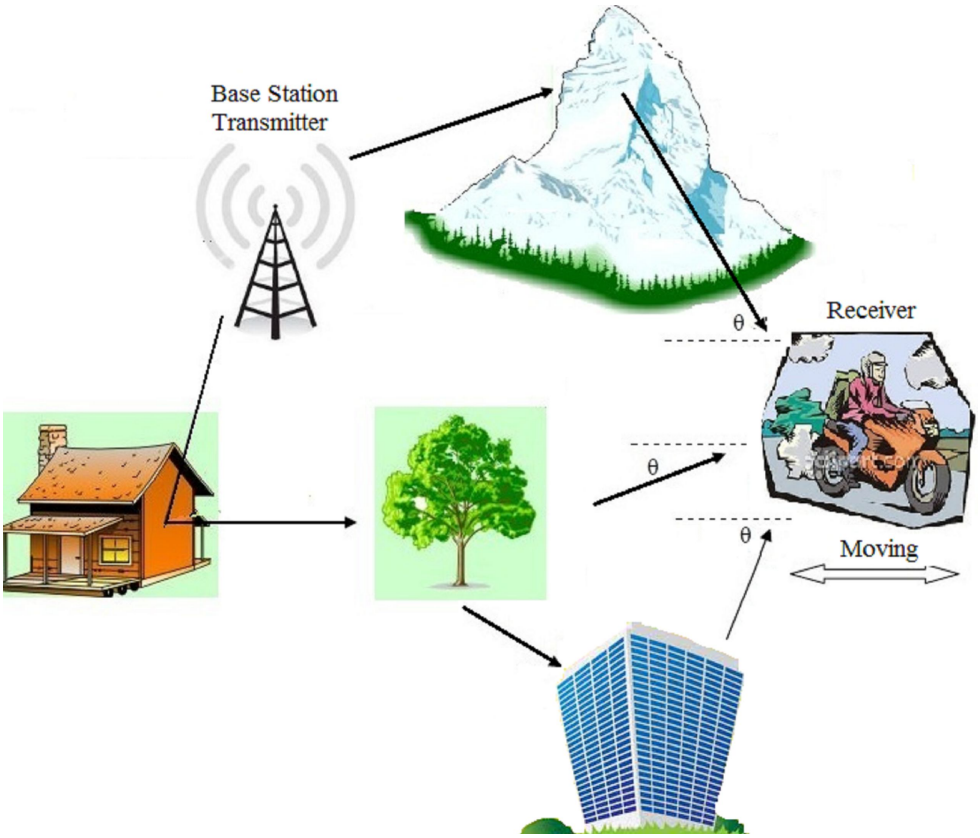


Figure 2.3: Signal propagates through a multipath channel.

2.3.1 Time Selective Fading

Relative motion between the transmitter and the receiver causes a transmitted signal and its reflections to undergo different frequency shifts, which are known as Doppler shifts. A received signal transmitted at frequency f_c will experience a Doppler shift given by [57]

$$f_d = \frac{f_c v_m}{c} \cos \beta, \tag{2.8}$$

where β is the incident angle of the received signal with respect to the direction of the receiver's motion, v_m is the velocity of the receiver towards the transmitter and c is the speed of light. In a multipath channel the signal is spread over the frequency range

$$f_c \pm f_{d_{max}}, \tag{2.9}$$

where the maximum Doppler shift is calculated as

$$f_{d_{max}} = \frac{f_c v_m}{c}. \tag{2.10}$$

This effect is known as frequency dispersion or time selective fading [57, 58].

A channel is considered as a slow fading channel if the channel impulse response changes at a much slower rate than the transmitted baseband signal [58]. This means that the channel can be assumed to be static over one or more symbol intervals. Equivalently, the Doppler spread, which is a measure of the frequencies over which the Doppler spectrum is large enough to cause frequency dispersion, is much less than the bandwidth of the transmitted signal. Conversely, for a fast fading channel, the Doppler spread is comparable to or larger than the bandwidth of the transmitted signal. In the time domain, the coherence time of the channel, which is approximately equal to the reciprocal of the Doppler spread [58], is less than the transmitted signal's symbol period, T. Fast fading occurs at very low data rates or at high vehicle velocity. In general, if the product $f_{d_{max}} T$, called the normalized fade rate, approaches 0.01 or greater, a channel is referred to as fast fading. Conversely, if the normalized fade rate is close to zero, a channel is recognized as slow fading. For the special case when $f_{d_{max}} T = 0$, the channel response is stationary between the transmitter and receiver, and the channel is called a static channel. No fading occurs in a static channel and so AWGN is the only cause of signal corruption.

In the following, we assume the channel to be fast fading since the system considered in this thesis is for radio mobile communication, which can involve high velocity motion between transmitter and receiver. Consider the P25 Phase 1 radio system [2], which transmits 4800 symbols per second (T = 1/4800 s). The system operates at the 900 MHz band radio; with a standard mobile speed of $70kmh^{-1}$, the maximum Doppler frequency is

$$f_{d_{max}} = \frac{900 \times 10^6 \times (70 \times 10^3/3600)}{2.99 \times 10^8} = 58.5(Hz). \tag{2.11}$$

If the maximum Doppler shift is about 58 Hz, the normalized fade rate for P25 Phase 1 system is 58/4800 = 0.012 that is corresponding to a fast fading channel. This proves that our earlier assumption is reasonable and can be used for the remainder of this thesis.

2.3.2 Frequency Flat and Frequency Selective Fading

As mentioned earlier, there are often obstacles between a signal transmitter and its corresponding receiver; this causes multiple reflected copies of the transmitted signal to arrive at the receiver at different times and hence overlap each other. This effect is known as time dispersion, which results in frequency flat or frequency selective fading [58] of the transmitted signal. If all of the spectral components are similarly affected, the fading is frequency non-selective or flat. Conversely, if distinct spectral components experience different magnitudes of fading, the fading is frequency selective. The coherence bandwidth is a statistical measure of the range of frequencies over which the fading may be considered flat, and is inversely proportional to the time delay spread of the fading channel. Spectral components within the band are passed with approximately the same gain and change in phase [58].

Frequency selective fading happens when the bandwidth of the transmitted signal is higher than the coherence bandwidth of the channel. Transforming this into the time domain, we can see that a frequency selective signal will have a multipath delay spread that is greater than the symbol period. Different gains are experienced over the transmitted signal band resulting in a variable frequency response. This is characteristic of wideband systems or long range communication. Frequency selective fading induces *inter-symbol interference (ISI)*, which requires equalization to mitigate its effect. On the other hand, flat fading occurs when the symbol period is

much greater than the multipath delay spread of the channel. In flat fading conditions, channel fading causes negligible ISI because the signal bandwidth is narrow relative to the coherence bandwidth of the channel. Flat fading channels often experience deep fades, but the ISI distortion is usually negligible due to the small delay spread, making it much easier to decode. Figure 2.4 illustrates the frequency responses ($H(f)$) of flat and frequency selective channels relative to signal bandwidth (W) and coherent bandwidth (ΔW_c).

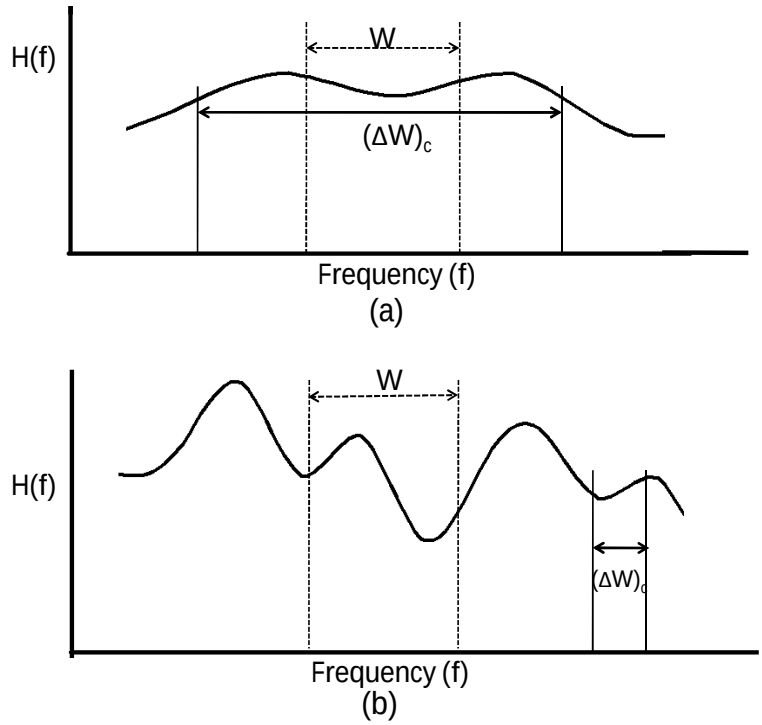


Figure 2.4: (a) Frequency flat channel (b) Frequency selective channel.

The predictive demodulator was originally designed to reduce the delay spread limitation of CPM reception without the need for equalization. Without any additional enhancement, the demodulator is capable of handling delay spread much higher than what is considered as flat fading [14]. However, for CPM system, the demodulator can not operate when delay spread exceeds $0.4T$, which is quite low for the channel to be considered as frequency selective [14]. Therefore, this research uses a generic multipath channel with considerable delay between the paths. The number of paths is chosen to be two since the second path can be considered to

be the sum of all the delayed signals relative to the first one. In most situations, the second arrival signal has weaker energy than the first arrival one because of its longer traveling distance; this has a natural effect of decreasing the interference in the received signal [58]. Here, to test the limitation of the system, it is assumed that the second path has the same average gain as that of the first path so the received signal can experience its worst distortion for a given delay spread.

2.3.3 Rician and Rayleigh Fading Channel

There are a large number of fading channel models that have been used in the literature, depending on the nature of the communication channel. The most commonly used models are Rayleigh, Nakagami and Rician channels because they are easy to analyze and are fairly realistic [59, 61]. The log normal fading model is typically used to describe shadowing and is almost always combined with other models. Typically, the Rician and Rayleigh fading channels are used to model land mobile radio channels. In urban environments, there are a large number of scatterers affecting propagation. If there is a dominant stationary (non-fading) signal component, such as a line of sight propagation, the Rician fading channel is an appropriate model. If there is no line of sight between the transmitter and the receiver, the Rayleigh fading channel may be used [60, 62].

In [62], the Rayleigh and Rician multipath fading channel can be represented by a discrete multipath model, for which the impulse response is

$$h_c(t,\tau) = \sum_{l=0}^{N_p-1} h_l(t)\delta(t-\tau_l). \tag{2.12}$$

Equation (2.12) shows a fading channel that consists of N_p paths, each with delay τ_l and complex gain $h_l(t)$ at observation time t, and $\delta(t)$ is the unit impulse response. The implementation assumes that the delay power profile and the Doppler spectrum of the channel are separable. This means that the complex path gains are uncorrelated with each other. In the case of a Rician channel, the fading process for each sub-channel is obtained as

$$h_l(t) = \sqrt{\sigma_l}\left[\frac{z_l}{\sqrt{K_{r,l}+1}} + \sqrt{\frac{K_{r,l}}{K_{r,l}+1}}\exp j(2\pi f_{d,LOS,l}t + \theta_{LOS,l})\right], \tag{2.13}$$

where σ_l^2 is the variance of $h_l(t)$, z_l is a complex white Gaussian random variable, $K_{r,l}$ is the Rician K-factor of the l^{th} path, $f_{d,LOS,l}$ and $\theta_{LOS,l}$ are the Doppler shift (in Hz) and initial phase (in rad) of the line of sight component of the l^{th} path.

The Rayleigh fading channel is a special case of the Rician channel with no line of sight component. For this model, the sub-channel gains are modeled as zero mean complex Gaussian random processes given by

$$h_l(t) = \sqrt{\sigma_l} z_l, \tag{2.14}$$

with variance σ_l^2 and normalized autocorrelation $R_h(t)$, which we can assume to be the Jakes autocorrelation function for isotropic scattering [63, 64]. The autocorrelation function of the l^{th} path gain can be expressed as

$$\begin{aligned} R_l(t) &= E\Big[h_l(t)h_l^*(\alpha - t)\Big] = \sigma_l^2 R_h(t) \\ &= \sigma_l^2 J_0(2\pi f_{d_{max}} t), \end{aligned} \tag{2.15}$$

where J_0 is the zero order Bessel function. In practice, there are often obstacles between transmitter and receiver in land mobile communication; hence, the presence of the line of sight path is quite rare. Therefore, in this research we model the channel as multipath Rayleigh fading.

Recalling that we assume the channel contains two independent paths ($N_p = 2$), a received signal can then be expressed as

$$r(t) = \sum_{l=0}^{1} h_l(t) s(t - \tau_l, \boldsymbol{a}) + n(t), \tag{2.16}$$

where $n(t)$ is the added white Gaussian noise from the receiver front end. Normally, the noise has a *power spectral density (PSD)* of N_0. However, if we use a unit power transmitted signal, which is equivalent to normalizing the received signal by $\sqrt{P}$ (the root of the transmitted power), the front end noise has a PSD of $N_0/P = N_0 T/E_s = T/\Upsilon$, where $\Upsilon = E_s/N_0$ is the symbol SNR.

2.4 Convolutional Code

The main convolutional code considered here is a rate 1/2 code that is used in the APCO-Phase 1, which is a well-known standard for public safety digital land

mobile radio. The standard is designed to provide interoperability for public safety professionals, as well as to enhance digital radio communication systems to achieve better spectrum efficiency, voice quality, user compatibility and system functionality [2]. The trellis for this code is shown in Fig. 2.5. Each state in this trellis is defined as a dibit. The convolutional code is a very simple one in which the current state is the last information dibit sent. Each state transition corresponds to a distinct output in the form of a pair of dibits. The state transitions and their corresponding encoded dibit pairs are shown in Table 2.1.

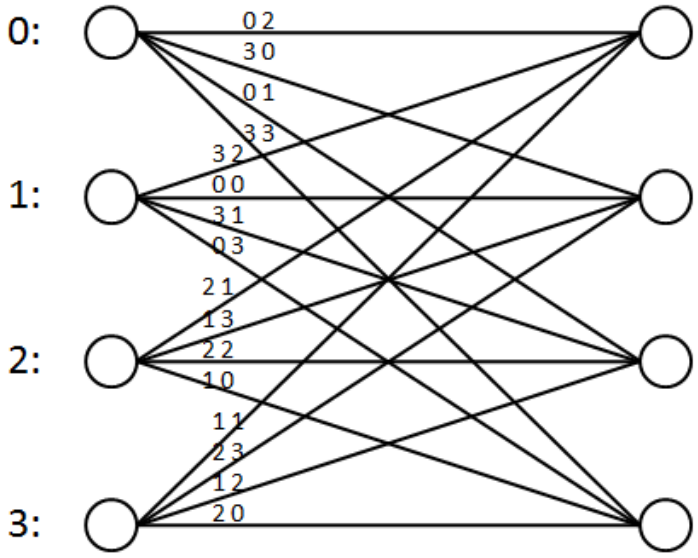


Figure 2.5: Half rate convolutional code trellis. Two dibits out on each branch.

Table 2.1: Rate 1/2 code dibit pairs output for different branches.

Branch	Input	Output	Branch	Input	Output
0-0	0	[0,2]	2-0	0	[2,1]
0-1	1	[3,0]	2-1	1	[1,3]
0-2	2	[0,1]	2-2	2	[2,2]
0-3	3	[3,3]	2-3	3	[1,0]
1-0	0	[3,2]	3-0	0	[1,1]
1-1	1	[0,0]	3-1	1	[2,3]
1-2	2	[3,1]	3-2	2	[1,2]
1-3	3	[0,3]	3-3	3	[2,0]

Figure 2.6 shows the shortest error event that can happen to this code. It can be seen that the shortest error event is two branches long, with all the paths separating from a common state, then passing through different intermediate states, and finally converging in a common destination state. In this case, to simplify our discussion, we consider the initial common state and the destination one are both state 0. Without loss of generality, we can assume that, among the paths in Fig. 2.6, the correct path is the one whose intermediate state is 0. When comparing this path with the others, it can be seen that the symbol (or dibits) sequence along the correct path is different to those of two alternative paths by 3 symbols; and it is different to the last one by all symbols. For example, the path whose middle state is 1 has the same last symbol of 2 with the correct path, and the one through state 2 has the correct first symbol, while the one through state 3 has all incorrect symbols. This pattern where two alternative paths differ to the correct path by 3 symbols holds for all choices of initial state and destination state as well as correct and erroneous paths.

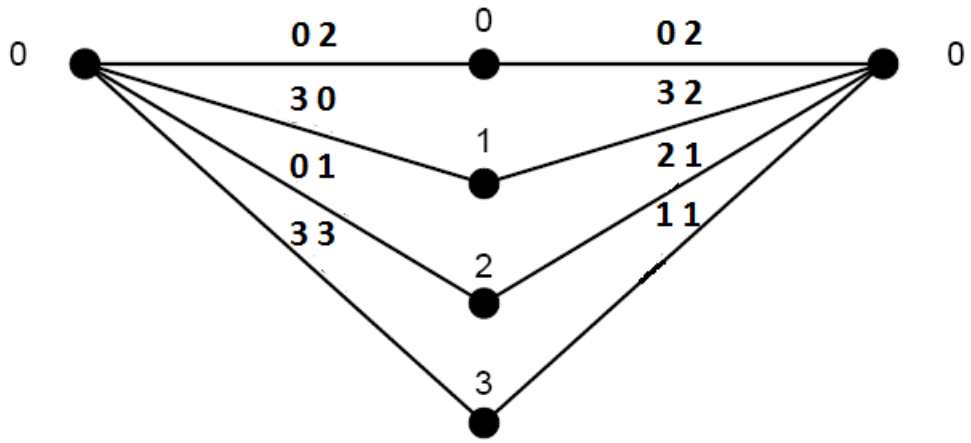


Figure 2.6: Shortest error event with beginning and ending states being 0.

Since there is an interleaver in this system, the result of having a difference of 3 symbols is the reduction in the potential order of temporal diversity by one in a fading channel. This loss in diversity is due to the fact that: The MLSE decoder chooses the path with the best metric (or the minimum Hamming distance) as the survivor path. The metric over the two branches of an error event is the sum of the metrics corresponding to the four symbols. When interleaving is applied to a long block, the four symbols can experience independent fades. Therefore, a code

designed for fading channel conditions is able to obtain fourth-order temporal diversity [8]. However, in this half-rate code, the correct path and alternative paths differ by only 3 symbols so the maximum temporal diversity order is three. This means that this code seems to be optimised for non-fading AWGN channels.

In addition to the rate 1/2 code, the APCO-Phase 1 standard also utilizes a rate 3/4 convolutional code, which is weaker [2]. However, it can still obtain dual temporal diversity. Like the rate 1/2 code, it has two dibits (two symbols) output per branch and its shortest error events have length of two branches. It has 8 states, which means each branch input is a tribit, and the current state is the last information tribit sent. The state transitions and their corresponding encoded dibit pairs are shown in Table 2.2.

Table 2.2: Rate 3/4 code dibit pairs output for different branches.

		Input Tribit							
		0	1	2	3	4	5	6	7
	0	0,2	3,1	3,2	0,1	1,3	2,0	2,3	1,0
	1	3,2	0,1	1,3	2,0	2,3	1,0	0,2	3,1
	2	2,2	1,1	1,2	2,1	3,3	0,0	0,3	3,0
Current	3	1,2	2,1	3,3	0,0	0,3	3,0	2,2	1,1
State	4	3,3	0,0	0,3	3,0	2,2	1,1	1,2	2,1
	5	0,3	3,0	2,2	1,1	1,2	2,1	3,3	0,0
	6	1,3	2,0	2,3	1,0	0,2	3,1	3,2	0,1
	7	2,3	1,0	0,2	3,1	3,2	0,1	1,3	2,0

Analysis of the rate 3/4 code distance properties shows that some competing paths in an error event differ in only two dibits. In those cases, the differing dibits are located on different branches, so the diversity order of two is preserved, even though interleaving is performed by a dibit pair [8]. To have a better performance analysis on the developed soft output algorithm, a number of simulations will involve both this and the rate 1/2 code. However, the latter is still the main code of interest.

Although both the trellis codes discussed here may not be very suitable for the considered channel, they are the only ECE codes applied in this system. This is because it is intended to verify the predictive detection scheme performance in a

system that has already been deployed so that we can find out how much gain, in practice, the techniques developed in this research offer. On the other hand, if accurate detection can be achieved with these not so good codes, then the application of better codes may provide better gains/improvements and thus, can be investigated in future work.

2.5 Interleaver

Interleaving is a common method to overcome error bursts due to correlated channel noise or fading. An interleaver is an input-output mapping device that permutes the ordering of a sequence of symbols from a fixed alphabet in a completely deterministic manner. It takes the symbols at the input and produces identical symbols at the output but in a different temporal order [56]. Thus, consecutive transmitted data are spaced apart before transmission. At the receiver end, the interleaved data is arranged back into the original sequence by the de-interleaver. As a result of the process, correlated noise introduced in the transmission channel appears to be statistically independent at the receiver. Yet, such a process also spreads up burst errors that are often caused by deep channel fades and thus allowing a better error correction. Interleaving can be considered as a form of time diversity. The interleaver can be of many types, of which the periodic and pseudo-random are two.

The interleaver used in this research is periodic and is the one used in the APCO Phase 1 standard. The interleaver operates on a data block whose length is 98 dibits long. Let $n_i = 8m + k$ be the input index of the interleaver, where m is the quotient and k is the remainder. For those input indices that have even values of k, there corresponding output indices n_o are computed as

$$n_o = \frac{n_i + 50k + c}{4}, \tag{2.17}$$

where $c = 0, 2, -4$ and -10 corresponds to $k = 0, 2, 4$ and 6. The output indices for the other values of n_i are

$$n_o = \frac{n_i - 1 + 50(k-1) + c}{4} + 1, \tag{2.18}$$

where now $c = 0, 2, -4$ and -10 corresponds to $k = 1, 3, 5$ and 7. The interleaving result of this design is shown in Table 2.3. Inspection also shows that it interleaves

Table 2.3: Interleaver table.

Index													
Out	In	Out	In	Out	In	Out	In	Out	In	Out	In	Out	In
0	0	14	56	28	10	42	66	56	28	70	84	84	46
1	1	15	57	29	11	43	67	57	29	71	85	85	87
2	8	16	64	30	18	44	74	58	36	72	92	86	54
3	9	17	65	31	19	45	75	59	37	73	93	87	55
4	16	18	72	32	26	46	82	60	44	74	6	88	62
5	17	19	73	33	27	47	83	61	45	75	7	89	63
6	24	20	80	34	34	48	90	62	52	76	14	90	70
7	25	21	81	35	35	49	91	63	53	77	15	91	71
8	32	22	88	36	42	50	4	64	60	78	22	92	78
9	33	23	89	37	43	51	5	65	61	79	23	93	79
10	40	24	96	38	50	52	12	66	68	80	30	94	86
11	41	25	97	39	51	53	13	67	69	81	31	95	87
12	48	26	2	40	58	54	20	68	76	82	38	96	94
13	49	27	3	41	59	55	21	69	77	83	39	97	95

by dibit pairs instead of by dibit. This means that both dibits on a branch of the rate 1/2 code have the same channel gain. This can have the effect of reducing the maximum temporal diversity order of the code from three (as discussed above) to two [8].

2.6 Noise-Limiting Filter

As shown in (2.16), a transmitted signal is not only affected by multipath fading but also degrades with the addition of white Gaussian noise. The effect of Gaussian noise can be reduced by the application of the noise limiting filter (referred to as the *front end filter (FEF)*), which reduces out of band noise as much as possible, while inflicting as little damage as possible on the desired signal. The FEF, denoted as $h_f(t)$, has a generally low-pass shape. In this research, we use an ideal rectangular low-pass, Blackman windowed FEF. It is a finite impulse response filter. The filter is normalized so that $h_f(t)$ has unit area,

$$\int_{-\infty}^{\infty} h_f(t)dt = 1. \tag{2.19}$$

Since Gaussian noise is included in the received signal, the FEF output, $y(t)$, consists of a signal component $x(t)$ and a noise component $\eta(t)$. From the received signal expression in (2.16), we have

$$\begin{aligned} y(t) &= x(t) + \eta(t) \\ &= \sum_{l=0}^{1} \int_{-\infty}^{\infty} h_f(\alpha) h_l(t-\alpha) s(t-\alpha-\tau_l, \boldsymbol{a}) d\alpha \\ &+ \int_{-\infty}^{\infty} h_f(\alpha) n(t-\alpha) d\alpha. \end{aligned} \tag{2.20}$$

The noise component $\eta(t)$ of $y(t)$ has the autocorrelation function

$$R_\eta(t) = \frac{T}{\Upsilon} \int_{-\infty}^{\infty} h_f(\alpha) h_f(\alpha - t) d\alpha = \frac{T}{\Upsilon} R_f(t), \tag{2.21}$$

where $R_f(t)$ is defined as the FEF impulse response autocorrelation and Υ is the SNR (as defined previously in Section 2.3).

2.7 Diversity Combining

Diversity combining is a technique that is used to increase the accuracy of signal reception in a wireless multipath fading channel. When radio channels are separated sufficiently in any of the space, frequency, time or polarization domains, each sub-channel experiences different fading conditions [44]. By combining the received signals from these sub-channels, a higher quality signal is obtained. Spatial diversity at the receiver is perhaps the most commonly used form of diversity. Its concept is relatively simple: Instead of using a single receive antenna, multiple receive antennas are deployed in such a way that the distance between any two of them is at least one wavelength. Because the antennas are spatially separated, the signals received by them typically experience different fading conditions. In practice, for typical mobile communication, the receive antennas must be separated by a fraction of the carrier wavelengths so that the correlation coefficients between the antennas are not so high. For example, 0.5 wavelength of separation can offer good deccorrelation depending upon the radio frequency environment that is being experienced. This means the fading experienced by these antennas can be considered as mutually independent [68]. This fading independence allows the received signals from different antennas to be combined so that the fading effect can be mitigated in the resulting

signal. Thus, the later process of information extraction can be done with greater accuracy. The three most common diversity combining techniques includes *selection combining(SC)*, *equal gain combining (EGC)*, and *maximal ratio combining (MRC)* [44].

It should be noted that signal combining is not the only method to utilize spatial diversity. Instead of combining the independent faded signals before extracting information, they can be jointly used right in the extraction process. Such diversity technique is commonly used in a number of innovation based receivers such as [74, 75] and the predictive receiver here. This is because it is quite straight forward to incorporate such diversity schemes into these receiver structures and there is no need to implement channel estimation. As a result of that, the diversity technique is the only one explored in [13, 14]. Although, satisfactory performance gain over a single antenna is achieved with it, the signal demodulation process can become quite computationally demanding. This happens because the demodulation takes into account all the independent received signals so its complexity grows near linearly with the number of receive antennas. Therefore, low complexity diversity combining techniques are of interest.

In any spatial diversity schemes at the receiver, the diversity gain increases with the number of receive antennas. However, in mobile communication, the number of antennas used is often restricted to 2 (dual diversity) or 3 (triple diversity) due to a number of practical reasons that includes complexity-performance trade off and signal independence. Dual and triple diversities have long been studied and implemented in real mobile communication systems [69, 70, 71]. It is well known that there is significant gain when a receiver changes from no diversity (a single antenna) to dual diversity. However, diminishing additional gains are achieved as the diversity order increases beyond two and higher order diversity is more complicated to implement in both hardware and software. In addition, because of the small size of the mobile device, it is quite difficult to keep the decorrelation coefficients small with a large number of antennas. Since low correlation is required for near independent fading, which is critical for diversity performance, the number of antennas on these devices are often less than four [93]. Because of the above restriction, this system considers the use of dual diversity as the main interest.

2.8 Summary

The preceding sections have outlined the communication system considered in this thesis. This includes the baseband transmission and reception models as well as the channel model used for simulation. Single-h partial response CPM modulation, in particular its C4FM variation, is the modulation of interest here. The interleaver and convolutional code are the other important parts in the transceiver system. Both the convolutional code and interleaver described in Section 2.4 and 2.5 are from the APCO Phase 1 standard. These may not be the best ones for the considered channel, but they allow us to investigate the advantages of the predictive detector in practical systems. It was also shown that the transmission channel can be modelled by a Rayleigh distributed, frequency selective fast fading channel with AWGN. This will affect the amplitude and the phase of the received signal, posing problems for the demodulation.

In the next chapter, the effects of these different aspects of the transmitter and the channel model on the receiver will be discussed in detail, leading to the actual design of the MLSE predictive receiver, and then to a reduced complexity version.

Chapter 3

Predictive Demodulator

3.1 Introduction

Typically the transmitted wireless signal is corrupted by AWGN and distorted by Rayleigh channel fading as shown in Chapter 2. As a result, the received signal will often be very different to the transmitted signal. To reliably obtain the transmitted information, both AWGN and fading must be compensated by the receiver. However, because of the random nature of the AWGN and the fading, the received signal will never be the same as the transmitted signal. Therefore, the demodulator must identify the most likely transmitted symbols according to some criteria of optimality based on the noisy, faded received signal [72]. For CPM signals, three common types of demodulation scheme are discriminator, differential and coherent demodulation. These demodulation methods perform well under AWGN, but may not be able to establish reliable detection when the channel exhibits dispersive fading with relatively high delay spread. Another form of CPM demodulation that can be found in the literature is linear predictive demodulation [73]. In contrast to the other techniques, this method can provide satisfactory accuracy in signal reception even for a severe channel that exhibits fast fading ($f_{d_{max}}T \geq 0.01$) and relatively long delay spread (up to half the symbol period) without additional equalization. Although possessing such an advantage, it has been traditionally not a common CPM reception technique due to its considerably higher complexity. However, due to recent advances in hardware, linear predictive receivers may become practical in real systems.

3. PREDICTIVE DEMODULATOR

This thesis considers a new linear predictive demodulator for CPM signal that can provide reliable signal detection under dispersive channel condition [13]. The application of linear prediction in signal demodulation is not a new topic. The first version of a predictive receiver for fading channels was proposed in [73]. It considered CPM on flat fading channels. Its extension to diversity reception and a receiver structure based on the VA with a LUT are shown in [74]. The first design of an explicitly predictive MLSE receiver for frequency selective fading is presented in [75]. It was extended to MAP reception in [76]. A receiver that combines linear prediction, the VA, *recursive least-squares (RLS)* estimation and a simplified model for selective fading is presented in [77]. This receiver differentiates itself from the others by having an adaptive prediction scheme that accommodates for the changes in Doppler spectrum and power delay profile.

Although the demodulator design here contains elements similar to some of the above linear prediction receivers, it also has some unique features. For example, it is different to the design of [73] in the prediction domain; its prediction is done on the FEF output signal while the prediction process of [73] operates on a modulation free signal (the demodulator removes the effect of modulation on the received signal by multiplying it with the conjugate of the hypothesis transmit signal). In addition, the receiver in [73] calculates its prediction coefficients directly from the autocorrelation of the received samples, while the receiver here computes its coefficient sets by reconstructing the expected received signal from a hypothesized transmitted symbol sequence based on the receiver knowledge of the channel statistics. The demodulator design is perhaps closest to the one in [75]; however, it differentiates itself from the latter by taking into account the effect of the noise limiting filter (FEF) on the received signal and having a pre-computed prediction coefficient LUT scheme for complexity reduction [13, 14].

In this chapter, the theoretical background and derivation behind the predictive demodulator structure in [13, 14] is reviewed. The chapter begins with an introduction to the system model. Following it, the derivation for MLSE detection and its metric is given. The construction of the demodulation trellis and the VA as a technique for implementing MLSE are then described for CPM signals. A trellis size reduction method for simplifying the demodulator operation is also included. After that, the receiver is extended to multiple samples per symbol and multiple

receive antennas are considered. The final section includes the computation of the predictive coefficients and the formation of the coefficient LUTs.

3.2 System Model

Figure 3.1 shows the baseband system model under consideration for the predictive demodulator. Here, we assume perfect synchronization.

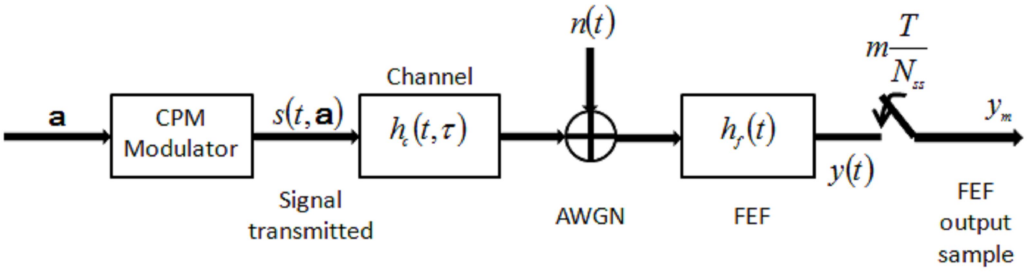


Figure 3.1: Pre-demodulation system model.

A sequence of encoded symbols $\boldsymbol{a} = (a_0, a_1, ..., a_{N-1})$ with a symbol period of T is modulated into a baseband CPM (C4FM) signal $s(t, \boldsymbol{a})$. We define a subsequence of $\boldsymbol{a}$ as $\boldsymbol{a}_j^n = (a_j, \ldots, a_n)$. In the n^{th} symbol interval, $[nT, (n+1)T]$, the transmitted signal, $s(t, \boldsymbol{a})$, may not only depend on the current symbol a_n but also on the previous symbol sequence, $\boldsymbol{a}_0^{n-1}$. In the case of a CPM signal, the number of prior symbols is determined by the effective phase response length, L, of the modulation.

The signal is transmitted over a mobile channel, which generally has a form of a Rayleigh multipath fading channel, with an impulse response, $h_c(t, \tau)$. The output of the channel is corrupted by AWGN, $n(t)$, with a PSD of $N_0/P = N_0T/E_s = T/\Upsilon$ (given that the signal power is normalized) where Υ is symbol SNR. The FEF limits the bandwidth of the received signal $r(t)$; by doing that, it colours the Gaussian noise contained in the signal. The channel and FEF can combine to form a composite filter with time-varying impulse response, $h(t, \tau)$. The equivalent system model resulting from such a combination is shown in Fig. 3.2. The signal at the FEF output can

then be expressed as

$$y(t) = \int_{-\infty}^{\infty} s(t-\tau)h(t,\tau)\,d\tau + \eta(t), \tag{3.1}$$

where $\eta(t)$ represents coloured Gaussian noise. The signal $y(t)$ is sampled N_{ss} times per symbol to produce a sample sequence $y_0, \ldots, y_{M-1}$, denoted $\boldsymbol{y}$ whose subsequences $y_j, \ldots, y_m$ are represented by $\boldsymbol{y}_j^m$.

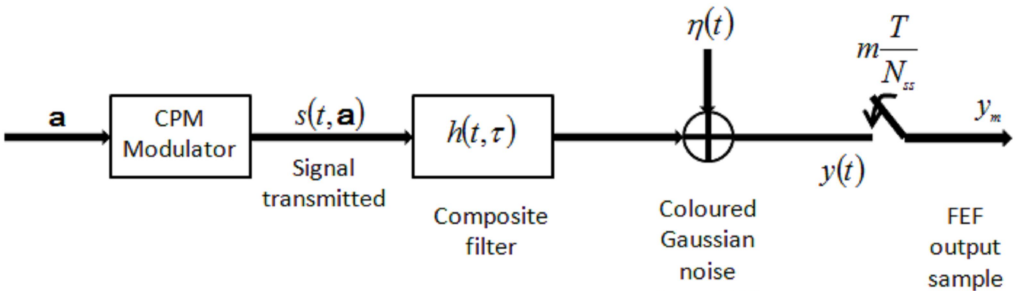


Figure 3.2: Equivalent pre-demodulation system model.

Before moving to the detection phase, there are two assumptions that we need to make:

- The fading channel impulse response is a Gaussian random process in time at any delay τ, due to the central limit theorem acting on signals received from many independent scatterers.
- The fading channel impulse response has a mean of zero and it is a Rayleigh fading channel.

From these assumptions, $y(t)$ in (3.1) is a zero-mean random process that is Gaussian when conditioned on the transmitted sequence $\boldsymbol{a}$ and has time-varying statistics. A channel with a line of sight component is not considered here because mobile communication does not usually have a line of sight path between transmitter and receiver. In most previous treatments, the receivers assumed the additive noise to be white and the FEF to be an ideal bandpass filter, which is wide enough to accommodate the received signal, even after fading, and the sampling to be at the Nyquist rate for the filter output. Although in principle this is the optimum filter and sampling rate [78, 79], there are many departures from that structure in

practice. The present design is the first to account for the more realistic situations of coloured additive noise and an arbitrary FEF in its derivation of the subsequent decision structure [13].

3.3 MLSE Detection and Metric

The predictive receiver in [13, 14] operates on the principle of MLSE. It chooses the sequence $\boldsymbol{a}$ that provides the highest probability $P(\boldsymbol{y}|\boldsymbol{a})$. Such a probability can be computed in recursive form as

$$P(\boldsymbol{y}|\boldsymbol{a}) = P(y_{M-1}|\boldsymbol{y}_0^{M-2}, \boldsymbol{a})P(y_{M-2}|\boldsymbol{y}_0^{M-3}, \boldsymbol{a}) \dots P(y_0|\boldsymbol{a}) \tag{3.2}$$

after the application of probability chain rule. This is equivalent to

$$\ln(P(\boldsymbol{y}|\boldsymbol{a})) = \sum_{m=0}^{M-1} \ln(P(y_m|\boldsymbol{y}_0^{m-1}, \boldsymbol{a})). \tag{3.3}$$

Since there can be multiple samples per symbol and the samples are assumed to be independent, the log-likelihood or detection metric can be further expressed as

$$\ln(P(\boldsymbol{y}|\boldsymbol{a})) = \sum_{n=0}^{N-1} \left(\sum_{j=0}^{N_{ss}-1} \ln(P(y_{nN_{ss}+j}|\boldsymbol{y}_0^{nN_{ss}+j-1}, \boldsymbol{a})) \right). \tag{3.4}$$

This equation means that the metric for the whole sequence is a sum of the metrics at successive symbol times, and each individual symbol metric is a sum of N_{ss} sample metrics.

Since the fading channel response is assumed to be a Gaussian random process, we can say that $\boldsymbol{y}$ is Gaussian when conditioned on the data sequence $\boldsymbol{a}$. Hence, each factor in (3.2) can be written as

$$P(y_m|\boldsymbol{y}_0^{m-1}, \boldsymbol{a}) = \frac{1}{2\pi\sigma_m^2(\boldsymbol{a})} \exp\left(-\frac{1}{\sigma_m^2(\boldsymbol{a})} \left| y_m - \hat{y}_m(\boldsymbol{y}_0^{m-1}, \boldsymbol{a}) \right|^2 \right), \ m = 0, 1, ..., M-1, \tag{3.5}$$

where $\hat{y}_m(\boldsymbol{y}_0^{m-1}, \boldsymbol{a})$ is the conditional mean (or the *minimum mean square error (MMSE)* prediction) of y_m, and $\sigma_m^2(\boldsymbol{a})$ is the prediction error variance [55, 56]. The

log-likelihood sample metrics in (3.4) are then

$$\begin{aligned}\ln(P(y_m|\boldsymbol{y}_0^{m-1},\boldsymbol{a})) &= -\ln(\sigma_m^2(\boldsymbol{a})) - \frac{\left|y_m - \hat{y}_m(\boldsymbol{y}_0^{m-1},\boldsymbol{a})\right|^2}{\sigma_m^2(\boldsymbol{a})} \\ &= -\ln(\sigma_m^2(\boldsymbol{a})) - \frac{|e_m|^2}{\sigma_m^2(\boldsymbol{a})}, \end{aligned} \tag{3.6}$$

where

$$e_m = y_m - \hat{y}_m(\boldsymbol{y}_0^{m-1},\boldsymbol{a}), \tag{3.7}$$

is the prediction error, which is also known as the innovations process [66].

The prediction $\hat{y}_m(\boldsymbol{y}_0^{m-1},\boldsymbol{a})$ is currently based on all the previous received samples. This is impractical since it would require very large memory and computational power. Therefore, we may assume that the significant part of the memory required for the prediction only extends K previous samples back in time (which is usually true). This means that y_m is predicted based on K previous samples, $\hat{y}_m(\boldsymbol{y}_{m-K}^{m-1},\boldsymbol{a})$. It should be noted that the actual number of samples used for the prediction process is K only when the sampling rate equals the symbol rate ($N_{ss} = 1$). Otherwise, the total number of samples used for the prediction is $KN_{ss} - 1$.

Since the sample sequence $\boldsymbol{y}$ is conditionally Gaussian, the predictor can be implemented as a data-dependent *finite impulse response (FIR)* filter that has K taps, $\boldsymbol{w}$. The prediction can then be computed as

$$\hat{y}_m(\boldsymbol{y}_{m-K}^{m-1},\boldsymbol{a}) = \boldsymbol{w}^\dagger(\boldsymbol{a})\boldsymbol{y}_{m-K}^{m-1}, \tag{3.8}$$

where the notation † corresponds to the complex conjugate operation. If we define the *prediction error filter (PEF)* coefficients to be

$$\tilde{\boldsymbol{w}} = \begin{bmatrix} 1 \\ -\boldsymbol{w} \end{bmatrix} = \begin{bmatrix} 1 \\ -w_0 \\ \vdots \\ -w_{K-1} \end{bmatrix} = \begin{bmatrix} \tilde{w}_0 \\ \tilde{w}_1 \\ \vdots \\ \tilde{w}_K \end{bmatrix}, \tag{3.9}$$

then the prediction error is given by

$$e_m = \tilde{\boldsymbol{w}}^\dagger(\boldsymbol{a})\boldsymbol{y}_{m-K}^{m}. \tag{3.10}$$

The coefficient set $\tilde{\boldsymbol{w}}$ can be computed from the Yule-Walker equation [60]

$$\boldsymbol{w}(\boldsymbol{a}) = \boldsymbol{R}(\boldsymbol{a})^{-1}\boldsymbol{p}(\boldsymbol{a}), \tag{3.11}$$

where $\boldsymbol{R}(\boldsymbol{a})$ and $\boldsymbol{p}(\boldsymbol{a})$ are the conditional covariance matrix and the conditional correlation vector of the sample sequence $\boldsymbol{y}_{m-K}^{m-1}$. These are given by

$$\boldsymbol{R}(\boldsymbol{a}) = E\left[\boldsymbol{y}_{m-K}^{m-1}{\boldsymbol{y}_{m-K}^{m-1}}^{\dagger}|\boldsymbol{a}\right] \tag{3.12}$$

$$\boldsymbol{p}(\boldsymbol{a}) = E\left[y_m^*\boldsymbol{y}_{m-K}^{m-1}|\boldsymbol{a}\right] \tag{3.13}$$

The values of the matrix and vector are dependent on the nature of the transmitted signal ($s(t,\boldsymbol{a})$), the channel ($h_c(t,\tau)$), noise effects ($n(t)$), the FEF characteristics ($h_f(t)$) and the sampling rate (N_{ss}). In other words, each coefficient set can be computed from a corresponding hypothesized transmitted symbol sequence, the channel and FEF models. This allows the computation for all the possible coefficient sets to be done during the receiver construction. Equations (3.12) and (3.13) are the most general formulas to calculate predictive coefficients; these formulations will be developed in detail specifically for our system in Section 3.4.

By examining (3.12), it can be seen that if $s(t,\boldsymbol{a})$ is replaced by an arbitrarily rotated version of $e^{j\phi}s(t,\boldsymbol{a})$, then the values of $\boldsymbol{R}(\boldsymbol{a})$ do not change. This is the classic phase ambiguity, which occurs in all phase-coherent systems. Because of it the receiver is unable to distinguish between different values of absolute phase of the sequence. Hence, the technique only works for systems, which modulate data into transmitted signal as phase differences such as *differential phase shift keying (DPSK)*, or CPM. For a CPM signal, the phase state is not important; only the correlative state affects the prediction process.

3.4 Detection by Trellis and Viterbi Algorithm

From (3.4), (3.6), and (3.8), the sequential log-likelihood can be computed for any given sequence $\boldsymbol{a}$. However, the detection process, which involves the calculations of the likelihoods for every single possible sequence, is not very practical. This is because the computation of the PEFs for each possible sequence is currently influenced by all the past symbols in the sequence. Fortunately, many modulation

methods allow a significant reduction in such dependence. In addition, the received sample sequence can be considered as the output of a hidden Markov model, which is a statistical Markov process with unobserved (hidden) states [80]. This allows the use of the Viterbi algorithm for signal detection.

For most modulation schemes, each sample of a modulated signal depends on a finite number of previous data symbols in a moving window fashion. For example, the frequency pulse of a CPM signal is approximately limited to a finite number of symbol times. It means that a signal sample is only affected by that number of adjacent symbols back in time. This leads [13, 14] to making another approximation that simplifies the receiver operation. Even after passing the signal through a frequency selective channel, the direct influence of any data symbol on a sample at the FEF output is limited to L symbol times, where L is the modulation phase response length. Given that the channel delay spread considered in this research is no more than half the symbol period, such an approximation should be sufficient. To illustrate this, we consider the effect of a_0, for example, on the FEF output signal; this effect starts from a near zero value at $t = 0$ and it ends at a value of zero, at $t = LT$. Figure 3.3 shows its span of direct influence for $L = 3$ and $N_{ss} = 2$.

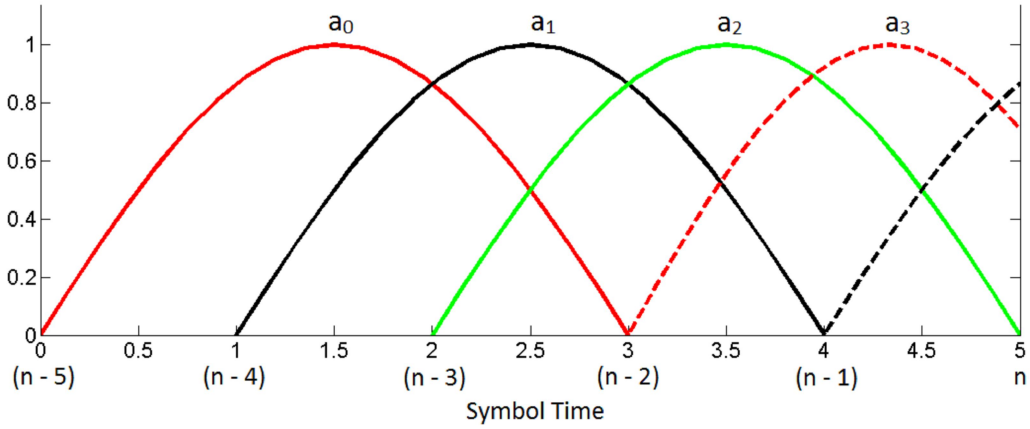


Figure 3.3: Historic symbols influence. Shown for $L = 3$, and $N_{ss} = 2$

Figure 3.3 indicates that any sample of the filtered received signal sample y_m only depends on $L = 3$ previously transmitted symbols. However, this dependence varies between the samples taken at integer symbol times, and at other times.

- A sample y_m in the interval $((n-1)T, nT)$ is influenced by the symbol sequence $\boldsymbol{a}_{n-L}^{n-1}$. For example, the sample at $t = 4.5$ is affected by $\boldsymbol{a}_2^4$.
- Sample y_m at integer time depends on only $L-1$ transmitted symbols, $\boldsymbol{a}_{n-L+1}^{n-1}$. For example, the sample at $t = 4$ depends only on $\boldsymbol{a}_2^3$.

The reduction in the number of involved symbols for samples at integer times is because each of the pulse shapes shown in Fig. 3.3 is zero at its right hand end.

To simplify the next part of the analysis, we consider the case of $N_{ss} = 1$ in which samples are taken at integer times, nT, so the sample index m is now equivalent to the symbol index n. From the previous section, we see that the sequence metric is computed from its corresponding symbol metrics (3.3), which is decided by a single sample metric of the form shown in (3.6). In (3.8), the sample metric is calculated via the PEF, which depends on the covariances calculated in (3.12) and (3.13) among $K+1$ consecutive samples of $y(t)$ (including the current sample that is being predicted). The prediction of sample $y(nT)$ is based on the K symbol spaced-samples $y((n-K)T)$ up to $y((n-1)T)$. Since each of these prediction basics is dependent on an overlap subset of $L-1$ data symbols, the prediction process involves a total of $L+K-1$ successive symbols $\boldsymbol{a}_{n-L-K+1}^{n-1}$. At this point, the complexity of the predictive receiver has been significantly reduced as the computation for the PEF and the prediction metric no longer needs to take into account the effect of any symbols that are $K+L$ symbol periods or more away from the current sample. This means that the calculations for the correlation matrix and vector in (3.12) and (3.13) respectively, can now be written as:

$$\boldsymbol{R}(\boldsymbol{a}) = E\left[\boldsymbol{y}_{n-K}^{n-1}\boldsymbol{y}_{n-K}^{n-1\ \dagger}|\boldsymbol{a}_{n-L-K+1}^{n-1}\right] \tag{3.14}$$

$$\boldsymbol{p}(\boldsymbol{a}) = E\left[y_n^*\boldsymbol{y}_{n-K}^{n-1}|\boldsymbol{a}_{n-L-K+1}^{n-1}\right] \tag{3.15}$$

It should be noted here that the matrix $\boldsymbol{R}(\boldsymbol{a})$ has size $K \times K$, while the vector $\boldsymbol{p}(\boldsymbol{a})$ has size $K \times 1$.

Moreover, the detection process can now be done via a trellis and the VA. The trellis states are defined by all possible combinations of $L+K-2$ successive data symbols. For example, state Φ_{n-2} is the integer equivalent of the sequence $\boldsymbol{a}_{n-L-K+1}^{n-2}$.

Each trellis branch is defined by the symbol a_{n-1}, while its corresponding metric is determined by the symbol sequence $\boldsymbol{a}_{n-L-K+1}^{n-1}$. This is because the sequence contains all the symbols involved in the state transition $\Phi_{n-2} \rightarrow \Phi_{n-1}$, which is $\boldsymbol{a}_{n-L-K+1}^{n-2} \rightarrow \boldsymbol{a}_{n-L-K+2}^{n-1}$. Hence, each state transition uniquely specifies a symbol sequence that, in turn, defines the PEF for computing the branch metric (which is also the symbol metric in this case) according to (3.9) and (3.11). By substituting (3.10) into (3.6), the branch metric, denoted as $\mu_n\Big(\boldsymbol{a}_{n-L-K+1}^{n-1}\Big)$, is

$$\mu_n\Big(\boldsymbol{a}_{n-L-K+1}^{n-1}\Big) = -\ln\Big(P\Big(y_n|\boldsymbol{y}_{n-K}^{n-1}, \boldsymbol{a}_{n-L-K+1}^{n-1}\Big)\Big) \tag{3.16}$$

$$= \ln\Big(\sigma_n^2(\boldsymbol{a}_{n-L-K+1}^{n-1})\Big) + \frac{\Big|\tilde{\boldsymbol{w}}^{\dagger}(\boldsymbol{a}_{n-L-K+1}^{n-1})\boldsymbol{y}_{n-K}^{n}\Big|^2}{\sigma_n^2(\boldsymbol{a}_{n-L-K+1}^{n-1})}, \tag{3.17}$$

which has been negated, so we can deal with positive quantities.

The overall sequence metric, which is minimized by the right choice of $\boldsymbol{a}$, is the sum of the branch metrics

$$M(\boldsymbol{a}) = \sum_{n=0}^{N-1} \mu_n\Big(\boldsymbol{a}_{n-L-K+1}^{n-1}\Big). \tag{3.18}$$

For values of n less than $K + L - 1$, the symbol values are assumed to be known at the receiver. These symbol values can be the end part of the synchronization sequence that precedes the data block.

It should be noted that the branch metric (and path metric) computation is optimized when the predictive coefficients are computed directly from the current channel condition. However, this can only be achieved by having a channel estimation scheme at the receiver. Given that the computation of each coefficient set $\tilde{\boldsymbol{w}}$ is a demanding task, this together with the extra complexity of the channel estimation process may consume a lot of computational power, which may not be ideal for mobile devices. As suggested earlier, a sub-optimal solution for this is to calculate all the possible coefficient sets based on the receiver's best knowledge about the channel prior to transmission. These sets are then stored in a LUT implemented in the receiver memory. Hence, the demodulator can quickly pull out the suitable coefficient set when a branch metric calculation happens. This method of using a LUT allows a large reduction in computational complexity and it has also shown in

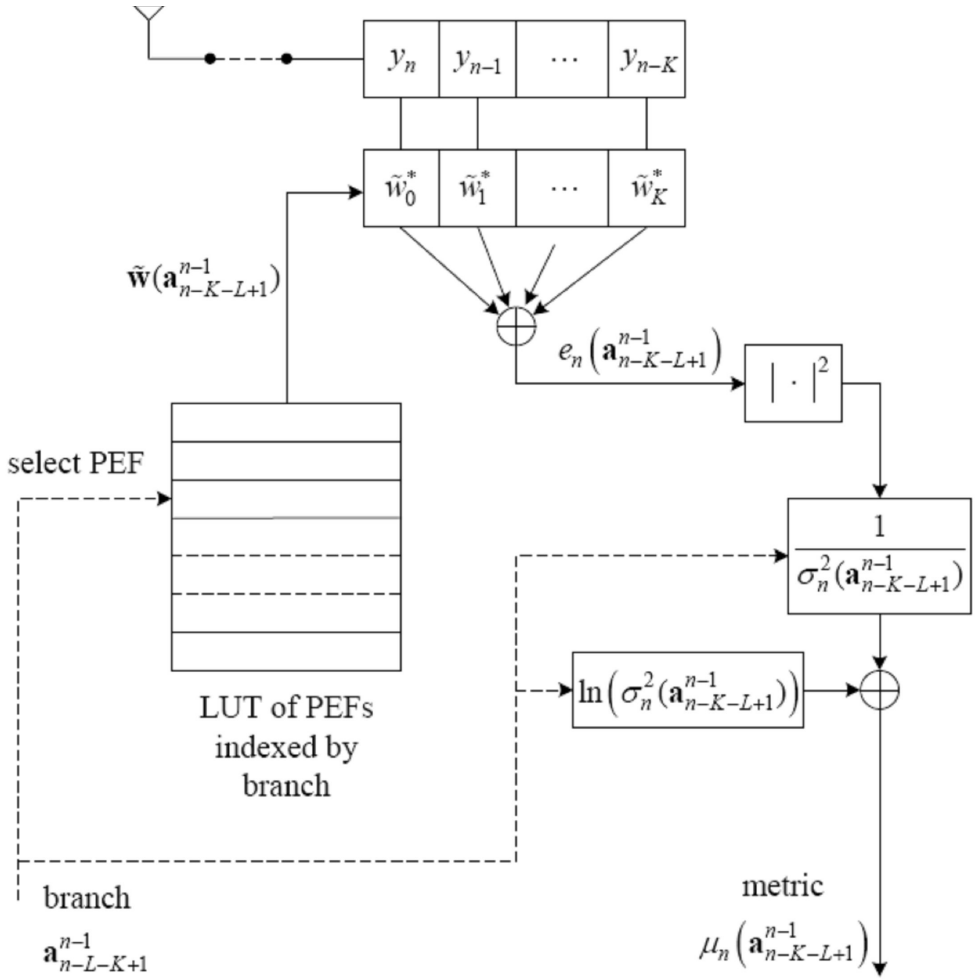


Figure 3.4: Branch metric calculator [13].

[14] to provide satisfactory performance. The branch metric calculation process can be summarized as in Fig. 3.4, where the LUT, as discussed earlier, represents the memory storage for all sets of PEF coefficients, $\tilde{\boldsymbol{w}}$, generated from all the possible sequences of $\boldsymbol{a}_{n-L-K+1}^{n-1}$.

With the above discussion, we can realize that the size of the trellis is determined by the constellation of the modulated signal, the phase response length L, and the prediction order K. If the constellation size is Q, then the trellis has

$$N_{state} = Q^{L+K-2} \tag{3.19}$$

states. The number of possible distinct branches is

$$N_{brn} = Q^{L+K-1}. \tag{3.20}$$

For example, if the constellation is binary, $L = 3$, and $K = 2$, then the trellis contains 8 states and there are 16 allowable transitions (or branches). Figure 3.5 shows this trellis for two contiguous symbol periods.

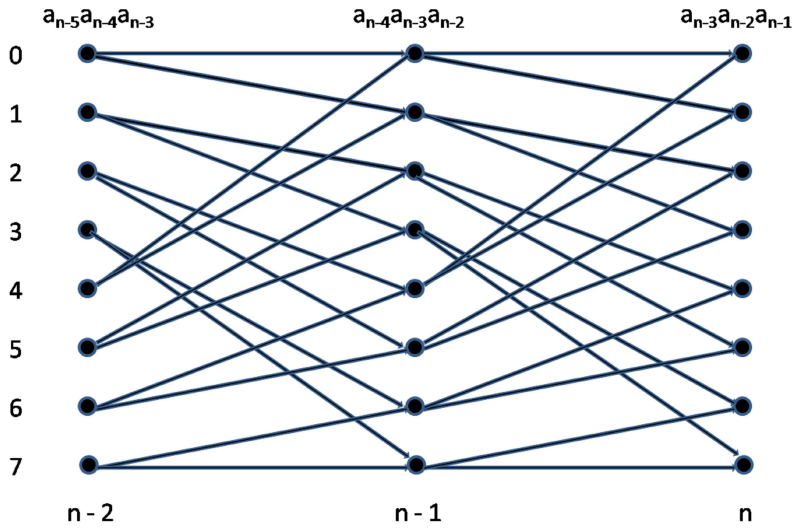


Figure 3.5: Trellis for binary symbols. Shown for $L = 3$ and $K = 2$.

3.5 Complexity Reduction Technique

From the previous section, we have seen how the trellis is formed from the modulation and prediction processes. According to (3.19), the trellis size is dependent on the prediction length K and the phase response length L. Simulation results in [14] show that the prediction needs to have $K = 3$ and $L = 3$ in order for the receiver to significantly outperform a differential receiver. Unfortunately, for $Q = 4$, this would mean that the corresponding trellis has $Q^{3+3-2} = 256$ states, which is very large for a number of practical implementations for mobile devices. Hence, additional methods must be applied to the receiver to reduce the number of trellis states. In [13, 14], such a method was found based on *per survivor processing (PSP)* [81].

The principle of PSP constitutes a general framework for the approximation of MLSE algorithms whenever the presence of unknown quantities prevents receivers from the realization of the classical VA [82]. In other words, PSP has a particular application whenever the transition metrics in the VA are affected by some degree of uncertainty that could be removed or reduced by data-aided estimation techniques. In a typical example, this uncertainty is often due to imperfect knowledge of some channel parameters, such as the carrier phase, timing epoch, or the channel impulse response itself. Therefore, the PSP technique is commonly applied to the areas of fading channel tracking [82], joint MLSE of data and channel [83, 84], and carrier phase synchronization for coded modulations [85, 86, 87]. Another particular application of PSP has appeared in the area of *reduced state sequence estimation (RSSE)* that is used to simplify receiver complexity for ISI channels [16, 88].

The application of PSP here is considerably different to those mentioned above as its focus is not on estimating an unknown channel parameter. Thus, this method may be considered as a minor or a different interpretation of the PSP principle. It is perhaps closest to RSSE in the sense that complexity reduction can be achieved by reducing the trellis size. In the case of RSSE, the trellis reduction is obtained by partial representation of the residual ISI constructed from the symbol estimates stored in the VA path history. Here, it is achieved by truncating the state to a shorter sequence of symbols, but keeps the remainder as a per-state decision feedback trail (or path) [13]. Each symbol sequence occupying a feedback trail can be found in the survivor path corresponding to the initial state of the feedback trail. The detailed operation of this reduction technique is described in the remainder of this section.

If we denote P as the reduction factor obtained with the PSP method then the new number of states in the demodulation trellis is given by

$$N_{state} = Q^{L+K-P-2}. \tag{3.21}$$

According to (3.21), if we use $P = 2$, the 256 state trellis (for $Q = 4$) can be shrunk to a trellis of 16 states as shown in Fig. 3.6. Each state in the new trellis is defined by a pair of symbols instead of a sequence of 4 symbols. Therefore, the number of possible states (or the trellis size) is significantly reduced.

We have not yet explained how the new 16 state trellis can replace the original 256 state one in the prediction process. Recall from the previous analysis that the prediction computation requires the symbol sequence, $\boldsymbol{a}_{n-L-K+1}^{n-1}$, which is $\boldsymbol{a}_{n-5}^{n-1}$ for this case. For the 256 state trellis, this sequence can be obtained by just going back to the previous state from the current state. This is because the current state (denoted as st) is defined as $\boldsymbol{a}_{n-4}^{n-1}$, and the previous state (denoted as pst) is $\boldsymbol{a}_{n-5}^{n-2}$. However, for the trellis in Fig. 3.6, such a method only allows us to have access from a_{n-3} to a_{n-1}. Therefore, if this trellis is applied for demodulation, we need to go back further in time to acquire the required sequence.

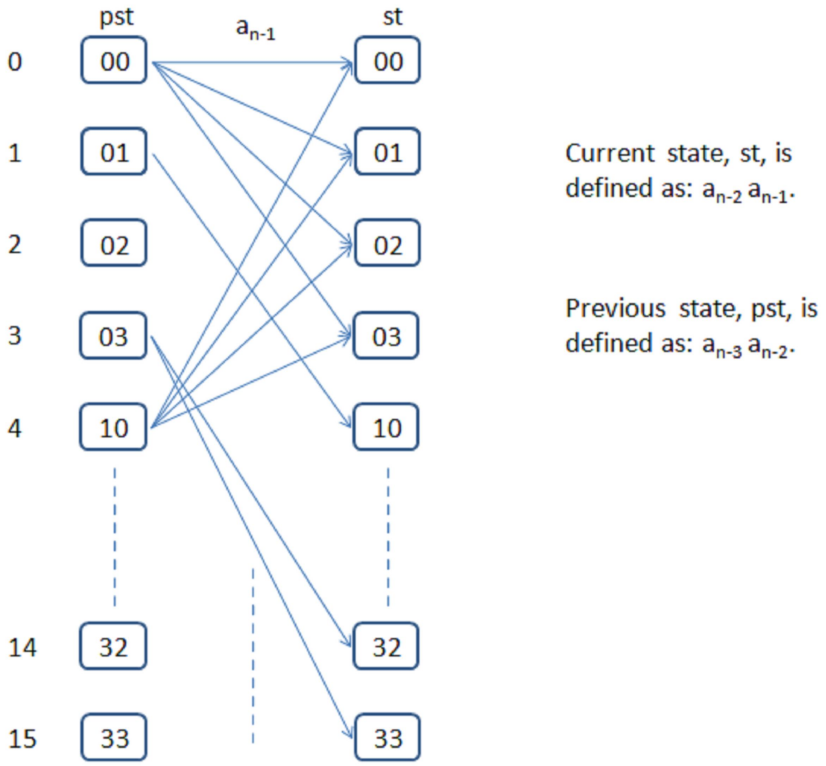


Figure 3.6: Reduced-size trellis with 16 States.

In this particular case, assuming the current state st is at time nT, we must go back to the past states as far as (and including) the state at time $(n-3)T$ (denoted as lst). At this stage, although the actual size of the trellis is reduced, there will not be a reduction in computation if we just simply trace all the possible paths back to

get all the combinations of the sequence $\boldsymbol{a}_{n-5}^{n-1}$. The application of PSP allows such a reduction to be done. The process stores the best branch leading to each state prior to st, then for the computation of the branch metric leading to st, we only need to consider the pst and the best path leading to it from the lst (or its feedback trail). The resulting sequence $\boldsymbol{a}_{n-5}^{n-1}$ is the most likely sequence corresponding to the transition metric between pst and st. Hence, the computation of the branch metric $[pst - st]$ can solely depend on such a sequence and can ignore all the other possible sequences that can be obtained by tracing back from pst to lst. It should be noted here that the first $L + K - 1$ symbols transmitted are known at the receiver so the branch metric computation for the early states must rely on these symbols.

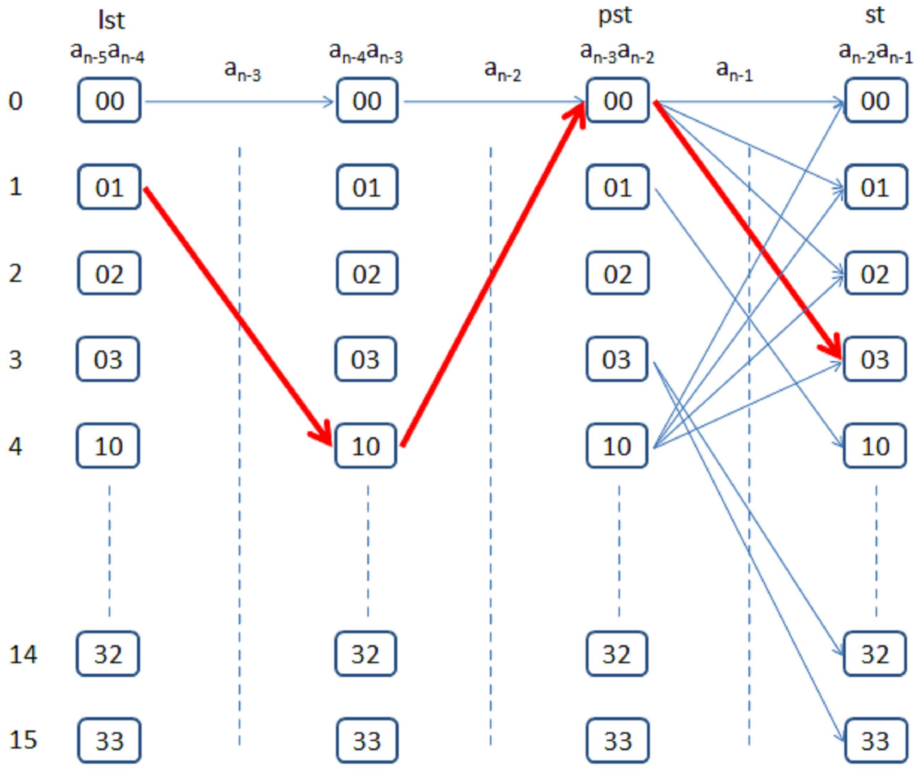


Figure 3.7: Prediction Process with Complexity Reduction Method.

The example in Fig. 3.7 shows the prediction made at state 3 ($st = 3$). Its previous state, state 0, gives us values for a_{n-2} and a_{n-3}. For each pst, we have saved the best trail leading to it. In the case of $pst = 0$, it is the path indicated by

the red arrows. The path contains state 4, and state 1 that consequently provide values for a_{n-5} and a_{n-4}. These together with the symbol a_{n-1}, which corresponds to the transition from $pst = 0$ to $st = 3$, make up the full (and most likely) sequence of $\boldsymbol{a}_{n-5}^{n-1}$ affecting the prediction. Thus, each time a branch metric is computed in the reduced-sized trellis, the demodulator has to take into account both the actual transition $[pst - st]$ and the best feedback path leading to pst instead of just the transition $[pst - st]$ in the case of the original trellis.

3.6 Multiple Samples per Symbols

In many modulation formats, such as CPM, the use of multiple samples per symbol can greatly improve the performance of their receivers. Hence, in this section, a modified design of the predictive receiver that can operate with multiple samples per symbol is developed.

For multiple samples per symbol operation, the number of involved symbols at any instant and the trellis size are the same as those of the single sample operation. In other words, these predictive parameters are independent of the number of samples per symbol. Hence, we can start by considering the familiar time line of the prediction for the single sample per symbol receiver as in Fig. 3.8.

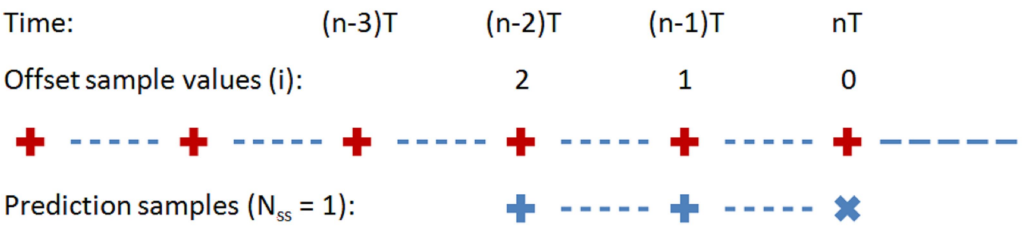


Figure 3.8: Time line for $N_{ss} = 1$ prediction of sample at nT with $K = 2$.

In Fig. 3.8, a sample is taken at every point marked by a red "+". These points are at times that are multiples of the symbol period T. To indicate each of these samples, we use an offset index i that counts back from nT. Hence, the corresponding sample times are $(n - i)T$. Below the time line are the samples involved in the prediction error calculation where a blue "+" represents a sample that the prediction

is based on, and a blue "×" represents the sample being predicted. For both single and multiple samples, the prediction process must obey the following:

- The samples involved in the prediction are those between $(n-K)T$ and nT inclusively.
- Each of the branch metrics, $\sigma_m\left(\boldsymbol{a}_{n-L-K+1}^{n-1}\right)$, in (3.18) is computed as the sum of the squared prediction errors (3.10) of the N_{ss} samples in the interval $((n-1)T, nT]$ according to (3.4).
- The prediction of each sample in the interval $((n-1)T, nT]$ depends on all the previous samples back to time $(n-K)T$, including those within the interval $((n-1)T, nT]$. This means that the predictions for different samples are based on different historic sample sequences. The sequence corresponding to the prediction for the latest sample is longest while that for the earliest sample is shortest. Hence, there are different PEFs for different samples being predicted within that interval.

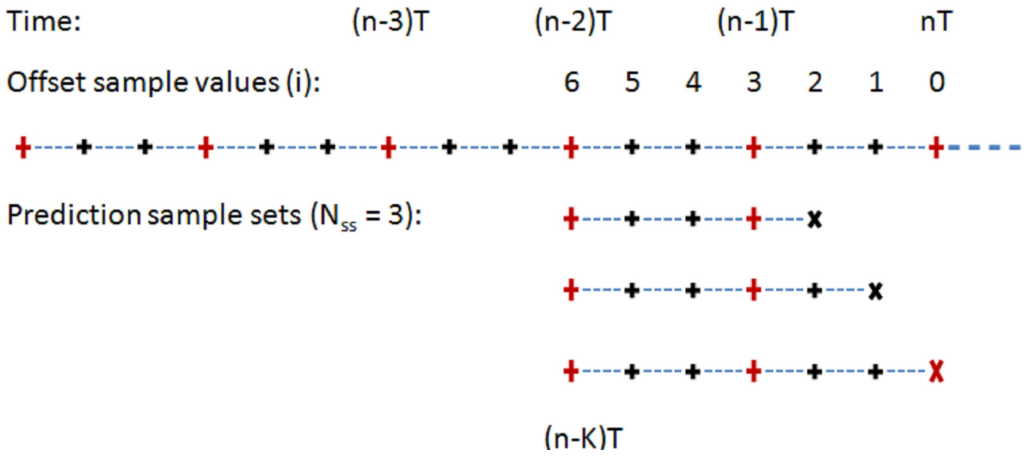


Figure 3.9: Time line for $N_{ss} = 3$ prediction of samples in $((n-1)T, nT]$ with $K = 2$.

Figure. 3.9 illustrates the timeline for the multi-sample scenario. Recalling that the offset index i counts back from the time nT, so the sample time is $(n - i/N_{ss})T$, or $(n - i/3)T$ in this example. Different sample sets for each of the predictions are shown below the timeline. We use the same notations to represent different types of samples as before, a "+" for the predictive samples, and a "×" for those being

predicted. From Fig. 3.9, it can be seen that the offset indices i of all the samples being predicted belong to the integer set $[0, 1, ..., N_{ss}-1]$. In general, for a prediction order of K, the prediction of a sample indicated by offset index i requires samples from $(n-(i+1)/N_{ss})T$ back to $(n-K)T$. For example:

- Prediction of the sample at $i = 0$ relies on samples from $(n-1/N_{ss})T$ back to $(n-K)T$ inclusively.
- Prediction of the sample at $i = 1$ relies on samples from $(n-2/N_{ss})T$ back to $(n-K)T$ inclusively.

  ......

- Prediction of the sample at $i = N_{ss}-1$ relies on samples from $(n-1)T$ back to $(n-K)T$ inclusively.

From the above analysis, we can see that the newest data symbol on which these predictions depend is a_{n-1}, and the oldest one is a_{n-K}. Hence, if the dependence between the data symbols is taken into account, then all the predictions rely on the symbol sequence, $\boldsymbol{a}_{n-K-L+1}^{n-1}$, which is exactly the same as that for the single sample case.

As the sample sequence required for each prediction of the sample in interval $((n-1)T, nT]$ has been specified, the corresponding correlation matrix (3.22) and vector (3.23) can now be constructed. Let's denote the sample with index offset i to be $y_{nN_{ss}-i}$ since all these samples are at times $(n-i/N_{ss})T$, then

$$\boldsymbol{R}_i(\boldsymbol{a}) = E\left[\boldsymbol{y}_{n-KN_{ss}}^{nN_{ss}-i-1}\boldsymbol{y}_{n-KN_{ss}}^{nN_{ss}-i-1\dagger}|\boldsymbol{a}_{n-L-K+1}^{n-1}\right] \tag{3.22}$$

$$\boldsymbol{p}_i(\boldsymbol{a}) = E\left[y_{nN_{ss}-i}^{*}\boldsymbol{y}_{n-KN_{ss}}^{nN_{ss}-i-1}|\boldsymbol{a}_{n-L-K+1}^{n-1}\right] \tag{3.23}$$

The matrix $\boldsymbol{R}_i(\boldsymbol{a})$ has size $(KN_{ss}-i)\times(KN_{ss}-i)$, while the vector $\boldsymbol{p}_i(\boldsymbol{a})$ has size $(KN_{ss}-i)\times 1$. The prediction filter and PEF for each sample being predicted are then

$$\boldsymbol{w}_i(\boldsymbol{a}) = \boldsymbol{R}_i(\boldsymbol{a})^{-1}\boldsymbol{p}_i(\boldsymbol{a}), \tag{3.24}$$

$$\tilde{\boldsymbol{w}}_i = \begin{bmatrix} 1 \\ -\boldsymbol{w}_i \end{bmatrix}. \tag{3.25}$$

Figure. 3.10 shows a block diagram of the multi-sample metric calculator for $K = 2$ and $N_{ss} = 2$. There are two "LUTs" corresponding to the two PEFs of the samples at time instant nT and $(n - \frac{1}{2})T$. The summer in each filter is hidden to reduce the complexity of the diagram. It should be noted here that since the two predictions are based on different sequences of past samples and these sequences are different in length, the PEFs differ in length and the corresponding variances $\sigma^2_{in}\left(\boldsymbol{a}^{n-1}_{n-L-K+1}\right)$, for $i = 0, 1$, are different. These variances must be applied separately to the outputs of the PEFs. It should be noted that although being different, these PEFs are all constructed based on the same channel conditions. Therefore, the coefficient LUTs corresponding to them can be grouped into a single LUT dedicated for such channel setting. This is also to avoid the confusion between the idea of using multiple LUTs with different channel settings in Chapter 4 and multiple PEF LUTs structure described here.

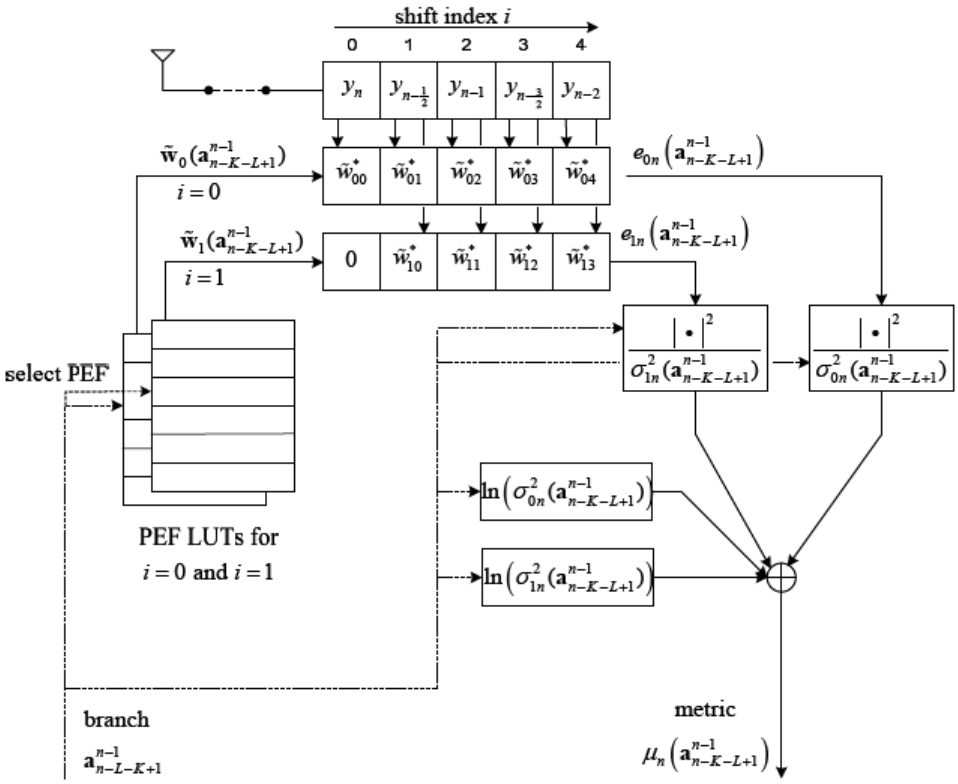


Figure 3.10: Multi-sample metric calculator for prediction of $K = 2$ with 2 samples per symbol, $N_{ss} = 2$ [13].

3.7 Spatial Diversity Reception

The benefits of diversity reception are well known. In this section, we describe the extension made to the predictive demodulation so that it can operate with diversity. Such an extension to diversity configurations can be done by simply summing the metrics calculated on the various diversity branches [13]. We assume that a transmitted signal is received on I antennas, each channel independent of the others, but with the same statistics (equivalently, with frequency diversity, the signal is received on I frequency branches). The receiver maximizes with respect to the data hypothesis $\boldsymbol{a}$, the likelihood

$$P(\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)}, \ldots, \boldsymbol{y}^{(I)}|\boldsymbol{a}) = P(\boldsymbol{y}^{(1)}|\boldsymbol{a})P(\boldsymbol{y}^{(2)}|\boldsymbol{a}) \ldots P(\boldsymbol{y}^{(I)}|\boldsymbol{a}), \tag{3.26}$$

where $\boldsymbol{y}^{(k)}$ is the vector of samples from diversity branch k. By taking the natural log to both side of (3.26), we have

$$\ln(P(\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)}, \ldots, \boldsymbol{y}^{(I)}|\boldsymbol{a})) = \sum_{k=1}^{I} \ln(P(\boldsymbol{y}^{(k)}|\boldsymbol{a})). \tag{3.27}$$

Each term on the right side of (3.27) can then be decomposed as in (3.3) to a sum of branch metrics, each a squared prediction error. Equivalently, it is possible to form a single branch metric as a sum of squared prediction errors (or branch metrics) across antennas (called *sum of squared prediction errors (SSPE)* diversity [13]). Following (3.6) and (3.17), the branch metric is given by

$$\mu_n\left(\boldsymbol{a}_{n-L-K+1}^{n-1}\right) = \sum_{k=1}^{I} \frac{\left|e_m^{(k)}\right|^2}{\sigma_m^2(\boldsymbol{a}_{n-L-K+1}^{n-1})} + \ln(\sigma_m^2(\boldsymbol{a}_{n-L-K+1}^{n-1})) \tag{3.28}$$

$$= \sum_{k=1}^{I} \frac{\left|\tilde{\boldsymbol{w}}^{\dagger}(\boldsymbol{a}_{n-L-K+1}^{n-1})\boldsymbol{y}_{n-K}^{(k)n}\right|^2}{\sigma_n^2(\boldsymbol{a}_{n-L-K+1}^{n-1})} + \ln\left(\sigma_n^2(\boldsymbol{a}_{n-L-K+1}^{n-1})\right). \tag{3.29}$$

Equation (3.29) looks complicated, but in operation it is not. Figure 3.11 illustrates that the same PEF is applied to each of the antenna outputs and the resulting squared errors are added together to form a state transition metric (the bias term $\ln(\sigma_m^2(\boldsymbol{a}_{n-L-K+1}^{n-1}))$ can also be added to the branch metric calculation, although experimentation shows that it makes little difference). As a result of that, the complexity of the receiver for implementing SSPE diversity rises almost linearly with the value of I as the branch metric computation can be considered as the

most demanding task in the demodulation process. Therefore, there is motivation to develop a suitable diversity technique that can provide good gain with low cost in computation.

3.8 Predictive Coefficient Generation

We recall that (3.11) gives the prediction error filters (PEFs) that are used in calculation of branch metrics in the VA. This section shows the development of computational expressions for the components of the arrays $\boldsymbol{R}(\boldsymbol{a})$ and $\boldsymbol{p}(\boldsymbol{a})$ in (3.11) for the case of C4FM and our channel model according to the predictive coefficient derivation in [13].

It can be seen that the components of the arrays $\boldsymbol{R}(\boldsymbol{a})$ and $\boldsymbol{p}(\boldsymbol{a})$ in (3.12) and (3.13) (and for the multi-sample version, the arrays $\boldsymbol{R}_i(\boldsymbol{a})$ and $\boldsymbol{p}_i(\boldsymbol{a})$ in (3.22) and (3.23)) are just covariances of the FEF output at different sample times. We represent such a component as

$$V_y(i_1, i_2|\boldsymbol{a}) = E\left[y\left(\left(n - \frac{i_1}{N_{ss}}\right)T\right)y^*\left(\left(n - \frac{i_2}{N_{ss}}\right)T\right)\right], \tag{3.30}$$

where i_1 and i_2 indicate sample times. The expectation is taken over the fading and noise ensembles, but not the data, since the result is intended to be a function of $K + L - 1$ symbols, $\boldsymbol{a}_{n-K-L+1}^{n-1}$. In detail, the arrays in the multi-sample case are then given, for $i = 0, \ldots, N_{ss} - 1$, by

$$\boldsymbol{R}_i(\boldsymbol{a}) = \begin{bmatrix} V_y(i+1, i+1|\boldsymbol{a}) & V_y(i+1, i+2|\boldsymbol{a}) & \ldots & V_y(i+1, KN_{ss}|\boldsymbol{a}) \\ V_y(i+2, i+1|\boldsymbol{a}) & V_y(i+2, i+2|\boldsymbol{a}) & \ldots & V_y(i+2, KN_{ss}|\boldsymbol{a}) \\ \vdots & \vdots & \ddots & \vdots \\ V_y(KN_{ss}, i+1|\boldsymbol{a}) & V_y(KN_{ss}, i+2|\boldsymbol{a}) & \ldots & V_y(KN_{ss}, KN_{ss}|\boldsymbol{a}) \end{bmatrix} \tag{3.31}$$

and

$$\boldsymbol{p}_i(\boldsymbol{a}) = \begin{bmatrix} V_y(i+1, i|\boldsymbol{a}) \\ V_y(i+2, i|\boldsymbol{a}) \\ \vdots \\ V_y(KN_{ss}, i|\boldsymbol{a}) \end{bmatrix}. \tag{3.32}$$

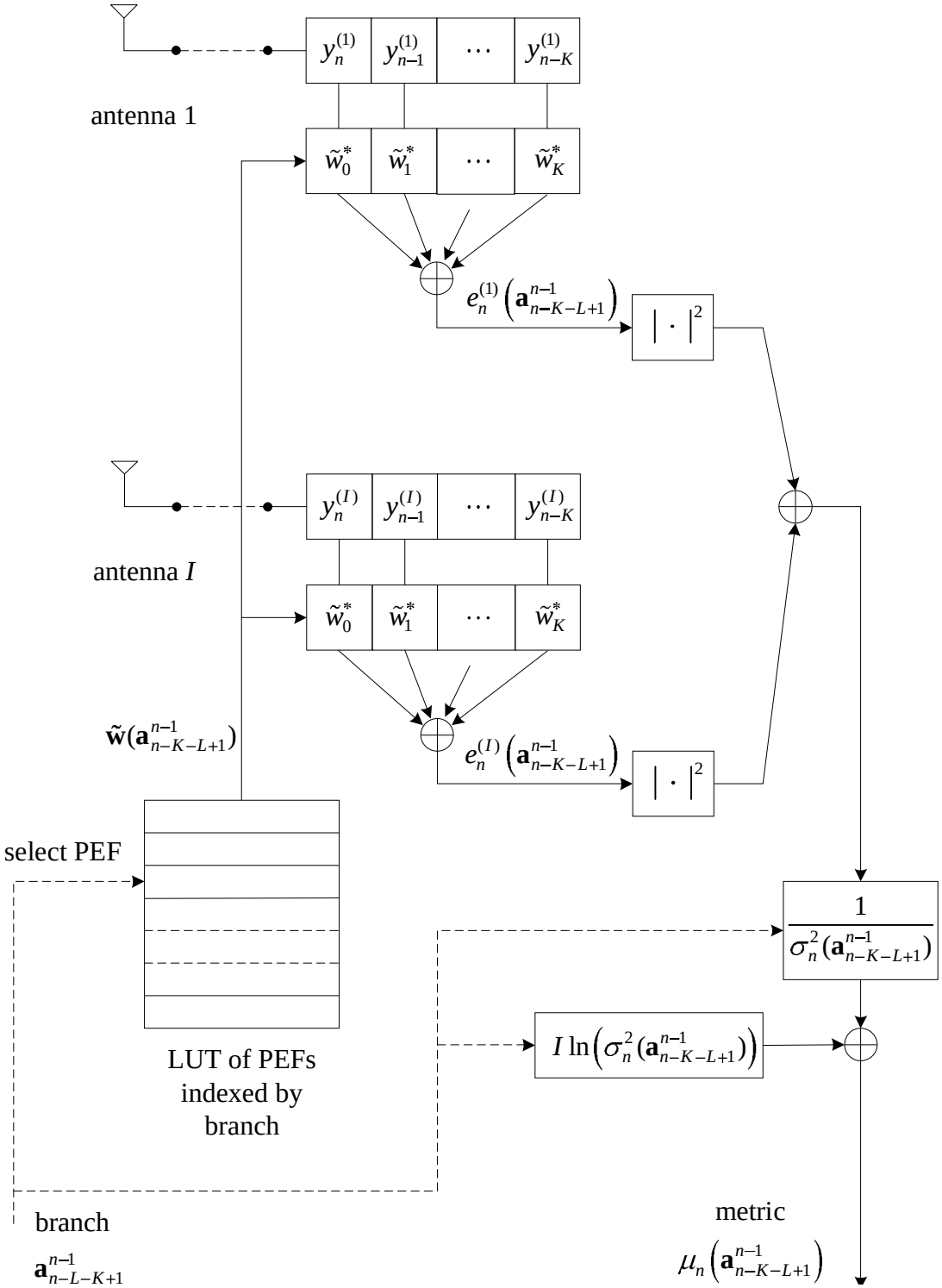


Figure 3.11: SSPE diversity implementation [13].

For the case of just one sample per symbol, N_{ss} is set to 1 and the arrays simplify.

In the following derivation, it is assumed that the selection of L for C4FM is generous enough that there is negligible ISI from data symbols outside the range $a_{n-L-K+1}$ to a_{n-1}. To start the derivation, the independence of the signal and noise terms in (2.20) makes the covariance (3.30) equal to

$$V_y(i_1, i_2|\boldsymbol{a}) = V_x(i_1, i_2|\boldsymbol{a}) + \frac{T}{\Upsilon} R_f\left(\frac{(i_2 - i_1)T}{N_{ss}}\right), \tag{3.33}$$

where $R_f(t)$ is defined in (2.21) (and is easily calculated), and $V_x(i_1, i_2|\boldsymbol{a})$ is defined as

$$V_x(i_1, i_2|\boldsymbol{a}) = E\left[x\left\{\left(n - \frac{i_1}{N_{ss}}\right)T\right\}x^*\left\{\left(n - \frac{i_1}{N_{ss}}\right)T\right)\right]. \tag{3.34}$$

The next goal is to obtain an expression for $V_x(i_1, i_2|\boldsymbol{a})$. Recall that $x(t)$ was defined in (2.20) as

$$x(t) = \sum_{l=0}^{1} \int_{-\infty}^{\infty} h_f(\alpha) h_l(t - \alpha) s(t - \alpha - \tau_l, \boldsymbol{a}) d\alpha, \tag{3.35}$$

where a computable expression for $s(t, \boldsymbol{a})$ over $[(n - K)T, nT]$ is defined as

$$s(t, \boldsymbol{a}) = \sqrt{\frac{E_s}{T}} \exp\left(j\phi(t, \boldsymbol{a})\right), \tag{3.36}$$

$$\phi(t, \boldsymbol{a}) = 2\pi h \sum_{k=n-K-L+1}^{n-1} a_k q(t - kT). \tag{3.37}$$

It should be noted that there is no phase state in (3.37) because as discussed before, the predictive receiver does not take into account the phase state of CPM. By substituting (3.35) into (3.34), we have

$$\begin{aligned} V_x(i_1, i_2|\boldsymbol{a}) &= \sum_{l_1=0}^{1}\sum_{l_2=0}^{1} \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} E\left[h_{l_1}\left\{\left(n - \frac{i_1}{N_{ss}}\right)T - \alpha_1\right\} h_{l_2}^*\left\{\left(n - \frac{i_2}{N_{ss}}\right)T - \alpha_2\right\}\right] \\ &\times s\left\{\left(n - \frac{i_1}{N_{ss}}\right)T - \alpha_1 - \tau_{l_1}, \boldsymbol{a}\right\} s^*\left\{\left(n - \frac{i_2}{N_{ss}}\right)T - \alpha_2 - \tau_{l_2}, \boldsymbol{a}\right\} \\ &\times h_f(\alpha_1) h_f^*(\alpha_2)\, d\alpha_1 d\alpha_2. \end{aligned} \tag{3.38}$$

Recalling that the complex gains of different channel paths are assumed to be uncorrelated, as in (2.15), so that the double sum in (3.39) collapses, this gives the more compact expression

$$\begin{aligned} V_x(i_1, i_2|\boldsymbol{a}) &= \sum_{l=0}^{1} \sigma_l^2 \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} R_h\left(\frac{(i_2-i_1)T}{N_{ss}} + \alpha_2 - \alpha_1\right) h_f(\alpha_1) h_f^*(\alpha_2) \\ &\times s\left\{\left(n - \frac{i_1}{N_{ss}}\right)T - \alpha_1 - \tau_{l_1}, \boldsymbol{a}\right\} \\ &\times s^*\left\{\left(n - \frac{i_2}{N_{ss}}\right)T - \alpha_2 - \tau_{l_2}, \boldsymbol{a}\right\} d\alpha_1 d\alpha_2. \end{aligned} \tag{3.39}$$

The expression for $V_x(i_1, i_2|\boldsymbol{a})$ in (3.39) can be further simplified by the following approximation: The channel fading is slow enough that the channel gain autocorrelation function $R_h(t)$ defined in (2.15) does not change significantly over the time span of the filter $h_f(t)$. This means that

$$R_h\left(\frac{(i_2-i_1)T}{N_{ss}} + \alpha_2 - \alpha_1\right) \approx R_h\left(\frac{(i_2-i_1)T}{N_{ss}}\right) = J_0\left(2\pi f_{d_{max}} \frac{(i_2-i_1)T}{N_{ss}}\right). \tag{3.40}$$

With the above approximation, the key formula (3.39) can be re-written as

$$\begin{aligned} V_x(i_1, i_2|\boldsymbol{a}) &= \sum_{l=0}^{1} \sigma_l^2 \int_{-\infty}^{\infty} s\left\{\left(n - \frac{i_1}{N_{ss}}\right)T - \alpha_1 - \tau_{l_1}, \boldsymbol{a}\right\} h_f(\alpha_1) d\alpha_1 \\ &\times \int_{-\infty}^{\infty} s^*\left\{\left(n - \frac{i_2}{N_{ss}}\right)T - \alpha_2 - \tau_{l_2}, \boldsymbol{a}\right\} h_f^*(\alpha_2) d\alpha_2 \\ &\times J_0\left(2\pi f_{d_{max}} \frac{(i_2-i_1)T}{N_{ss}}\right), \end{aligned} \tag{3.41}$$

where the zero order Bessel function is defined as

$$J_0(x) = 1 - \frac{(x/2)^2}{(1!)^2} + \frac{(x/2)^4}{(2!)^2} - \frac{(x/2)^6}{(3!)^2} + ... \tag{3.42}$$

As the functional forms of all factors in the left hand side of (3.41) are known apart from those associated with the transmission channel such as $f_{d_{max}}$, the calculation of $V_x(i_1, i_2|\boldsymbol{a})$ depends on the approximations of the channel factors. Given a certain set of approximation values for these unknown factors, the combination of (3.41)

and (3.33) allows all components of the arrays $\boldsymbol{R}(\boldsymbol{a})$ and $\boldsymbol{p}(\boldsymbol{a})$ to be computed. Based on (3.24), the resulting arrays are then applied to solve for the PEF $\tilde{\boldsymbol{w}}_i$ of a sequence $\boldsymbol{a}_{n-K-L+1}^{n}$ associated with a trellis branch. Thus, the quality of the PEF (or the branch metric) depends on the selected values of the channel factors.

The above method is used to obtain all the coefficient sets computed from all the possible combinations of $\boldsymbol{a}_{n-K-L+1}^{n}$. These coefficient sets are then stored in an LUT (or LUTs for the case of multi-sample per symbol), which is then embedded in the predictive receiver memory. Within a LUT, a coefficient set is indicated by its unique index, which is calculated from the sequence $\boldsymbol{a}_{n-K-L+1}^{n}$ corresponding to that set. The receiver relies on such an index to draw the correct coefficients from the LUT when computing branch metrics. As discussed earlier, the PEF calculation is greatly affected by a number of channel characteristics, which includes SNR (Υ), Doppler shift ($f_{d_{max}}$) and delay spread (τ_l). Since a coefficient LUT is calculated before the demodulator starts operating, those channel factors can only be estimated during the LUT construction. As shown in [14], the predictive demodulator performs best when these channel estimations are close to the actual channel and its performance degrades as these estimations are further away from their actual values. Given that this research considers mobile channels, which vary over time, it is impossible to guarantee that a single LUT constructed from a certain configuration will always be suitable for the current channel. A possible solution for this is to apply multiple LUTs whose channel factor configurations are different from one another so that the channel variation can be divided into different regions corresponding to the configurations. Thus, for a given actual channel condition, there is a much higher chance that it is close to one of the configurations than in the case of using only one configuration. Since each LUT can only be computed from one configuration, the use of multiple configurations corresponds to the use of multiple LUTs, which leads to multiple parallel prediction processes within the demodulation. In this thesis, the research on the use of multiple LUTs will be presented in Chapter 5.

3.9 Summary

In this chapter, a predictive MLSE C4FM receiver structure has been specified. In summary, the key points of the design include:

- MLSE detection is based on minimization of a sequence metric with respect to choice of the entire data sequence.
- For Rayleigh fading channels, a sequence metric consists of a sum of symbol metrics, each a squared prediction error (or sum of squared prediction errors, if multiple samples per symbol) from prediction of FEF output samples. The CPM phase state has no value on the computations of symbol metrics.
- For Rayleigh fading channels, linear prediction is optimum, and the PEF is obtained by solution of the Yule-Walker equation, which contains two arrays of covariances, both dependent on the data hypothesis. The prediction error can be based on a finite number K of successive symbol intervals, then the Yule-Walker equation and the PEFs are of finite size, and do not continue to grow with sequence length.
- The span of direct influence of any data symbol on the FEF output samples is limited to L symbol times, then any symbol metric depends on a moving window of data symbols. Therefore, the VA and PSP technique can be used to reduce computation significantly.
- Sampling more than once per symbol is sometimes essential to good performance. In this case, the branch metric is a sum of scaled, squared prediction errors, one for each of the samples in a symbol.
- The receivers structure can be based on pre-calculated LUTs and a simple structure to incorporate the LUTs into the branch metric computation process.

In the next chapter, this receiver structure will be extended with the application of soft output algorithms so that it is able to pass soft information to the subsequent decoder, and the overall system performance can then be increased.

Chapter 4

Reliability Generation and Iterative Detection

As discussed in Chapter 1, by letting the predictive demodulator feed reliability or soft information to the subsequent convolutional decoder, the overall receiver performance can be improved. There are several methods available for generating reliability information from the trellis operation. In this chapter, a low complexity SOVA-based algorithm is presented in detail. In addition, a higher complexity BCJR-based algorithm is derived specifically for this design to provide a performance comparison. Another focus of this chapter is to establish an iterative receiver structure, where soft information is passed between the predictive demodulator and the ECE decoder. Soft information exchange (iterative processing) achieves higher detection accuracy than that of the non-iterative system and the overall performance gain can increase with each iteration [89, 90]. Therefore, it is natural to take another step of extending the system here from soft detection to iterative detection, which can provide significantly better performance. The performance gains from these enhancements will be evaluated for different channel conditions. The results obtained are shown and discussed in the last section of this chapter.

4.1 Original SOVA

In this section, we summarize the original SOVA design for binary trellises and then discuss different complexity reduction methods.

4.1.1 Orignal SOVA

The original SOVA, as described in [28], was developed for a two stage receiver structure as shown in Fig. 4.1.

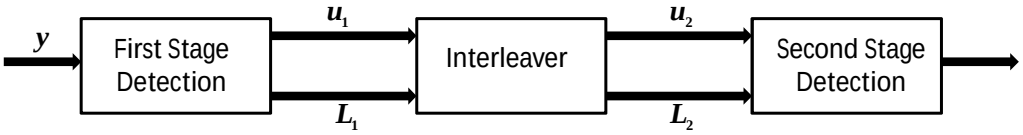


Figure 4.1: A general 2 stage detector model.

Here, the first stage can be a Viterbi demodulator while the second stage may be a Viterbi decoder. The application of SOVA in the demodulation process allows the first stage to not only provide hard decision symbols ($\boldsymbol{u}$), but also to deliver reliability information corresponding to these decisions ($\boldsymbol{L}$). The SOVA of [28] is restricted to a binary trellis with two branches ending in each node.

Assume that the VA in the first stage detection makes a final decision after a delay of δ_m, which is high enough that all 2^v survivor paths have been merged with sufficiently high probability (where v, in this case, is the memory of the modulation or convolutional code depending on whether the first detection stage is a demodulation or decoding process). Having said that, let us consider the case in which the VA in stage one has to select a survivor for state s_k at time kT as shown in Fig. 4.2. It selects the path with the smallest path metric to be the survivor. If we assume that the survivor path has index $m = 1$ and metric M_1, and the other path has metric $M_2 \geq M_1$, then the probability of selecting the wrong survivor path is

$$p_{sk} = \frac{e^{-M_2}}{e^{-M_2} + e^{-M_1}} = \frac{1}{1 + e^{M_2 - M_1}} = \frac{1}{1 + e^{\Delta}} \tag{4.1}$$

with $\Delta = M_2 - M_1 \geq 0$. From (4.1), we can see that p_{sk} approaches 0.5 if $M_1 \approx M_2$ and 0 if $M_1 \gg M_2$. With probability p_{sk}, the VA makes errors in all e positions where the information bits/symbols differ between path 1 and 2,

$$u_j^{(1)} \neq u_j^{(2)}, j = j_1, ..., j_e. \tag{4.2}$$

Positions where $u_j^{(1)} = u_j^{(2)}$ are not affected by the survivor decision.

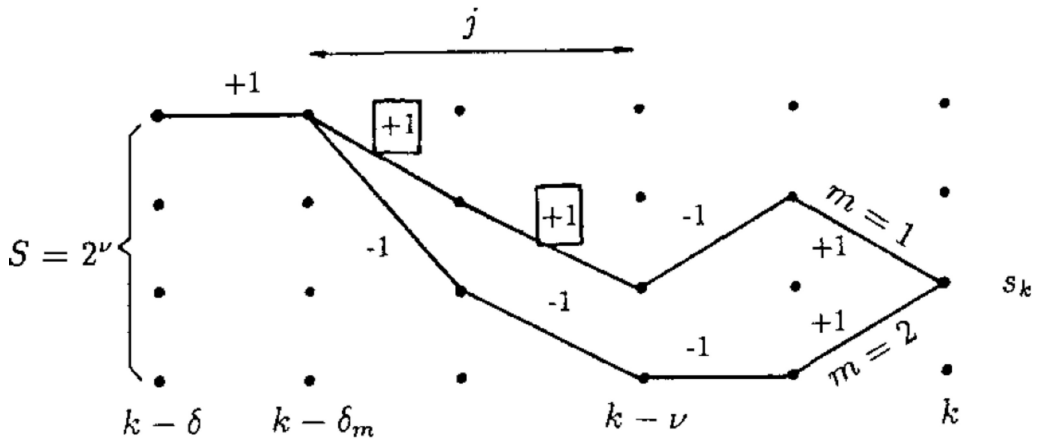


Figure 4.2: Example of the SOVA [28].

If we let δ_m be the length of those two paths until they merge, then we have e different information values and $\delta_m - e$ identical information values. Assume we have stored the probabilities $\hat{p}_j$ of previous erroneous decisions with path 1. We can update these probabilities with the newly computed p_{sk} for the e differing decisions on this path according to

$$\hat{p}_j = \hat{p}_j(1 - p_{sk}) + (1 - \hat{p}_j)p_{sk}, \;\; j = j_1, ..., j_e, \tag{4.3}$$

$0 \ll \hat{p}_j \ll 0.5$. This formula requires statistical independence between $\hat{p}_j$ and p_{sk}, which is true for most cases. The soft information, $\hat{L}_j$, can be computed via the log-likelihood ratio

$$\hat{L}_j = \frac{1}{\alpha} \log \frac{1 - \hat{p}_j}{\hat{p}_j}, \quad 0 \ll \hat{L}_j \ll \infty, \tag{4.4}$$

where α is a scaling factor that prevents overflow and allows $E\{\hat{L}_j\} = 1$. A good approximation for the above computation is [28]

$$f(\hat{L}_j, \Delta) = min(\hat{L}_j, \Delta/\alpha). \tag{4.5}$$

The process of updating the soft information can be summarized in the following:

For each state s_k
- Store $\Delta = max(M_m) - min(M_m)$.
- Initialize $\hat{L}_j(s_k) = +\infty$.
- For $j = k - v$ to $j = k - \delta_m$

Compare the two paths merging in s_k:
If $u_j^{(1)}(s_j) \neq u_j^{(2)}(s_j)$ then update
$\hat{L}_j = f(\hat{L}_j, \Delta)$.

The hard decisions ($\hat{u}_i$) and the soft information ($\hat{L}_i$), computed from the first detection stage's trellis via the SOVA, are used to calculate the soft decisions ($\hat{\Lambda}_i$) that will be fed to the VA of the second detection process. The soft decision is calculated according to

$$\hat{\Lambda}_i = \hat{u}_i \hat{L}_i, \tag{4.6}$$

where $\hat{u}_i$ has a hard value of 1 or -1.

4.1.2 Simplification Techniques

The sub-optimal SOVA is generally less complex than the optimal MAP algorithm. However, the computational demands of these algorithms for a large non-binary trellis are still very high. Since nowadays, most practical trellis applications are non-binary, many variations of SOVA employ different complexity-reducing techniques for simplifying their operations such as those in [32, 35, 36]. These simplifications are usually done by:

- Avoiding the use of complex mathematical operations such as log(), min() and max().
- Decreasing the length of the search-back/updating window (which is δ_m in the case of the original SOVA described previously).
- Reducing the number of paths involved in the calculation of reliability information.

It should be noted that each of these methods has a negative effect on performance. For example, in [30], the original SOVA is simplified by applying the first method where the complex selection operation (min() and max()) are approximated with the use of algebraic rings. This method significantly speeds up the SOVA decoder as it allows the use of parallel processing; however, an increase in BER is observed. The second simplification degrades performance by keeping the updating window as small as possible, thus preventing the past symbol reliability being updated by the latest information. The third method introduces performance degradation by

simply limiting the amount of information available for reliability calculation.

Depending on the characteristics of different applications, one (or a combination) of the above methods may be used. For the predictive demodulation trellis, it is realized that the search-back length is relatively short (2 to 3 symbols long); hence, the second simplification method may not be very useful here. My literature review has shown that the SOVA algorithm proposed by Turner in [29, 36] utilizes the remaining techniques very effectively. He considered the application of SOVA for a non-binary trellis for equalization in which there are M branches ending in each state. The complexity of such a process can be very high as the trellis is quite large. Fortunately, Turner discovered that reasonable SOVA performance can be achieved by looking only at the survivor path and the second best path at each survivor state. Furthermore, instead of computing the soft information as a log-likelihood ratio, Turner directly used the scaled path metric difference as the soft information. Turner's approach is a very attractive method to design a low-complexity soft output algorithm, but it can not be directly applied to the system here. This is because the algorithm was designed for channel equalization, which is able to produce meaningful soft symbols from its trellis operation even if only the two best paths are considered at each state. In contrast, the system considered here requires all the paths to produce meaningful soft symbols. As shown in the next section, a solution to this is to approximate symbol probabilities from the information available on the two paths, and then use these to compute soft bits. This, however, prevents the direct use of path metric difference as the symbol reliability measure.

4.2 SOVA-based Algorithm

We now describe the SOVA-based algorithm for the non-binary demodulation trellis. Two operations are involved, namely the approximation of the survivor symbol probability and the calculation of the corresponding bit reliabilities.

4.2.1 Symbol Probability Computation

The approximation for the statistical reliability of each symbol in the survivor path is described in the following: Consider Fig. 4.3 that shows a section of the demodulation trellis for a C4FM signal in which there are four paths entering each trellis

state. The survivor path is indicated by the red line (with diamond marker) while the blue line (with triangle marker) and green line (with circle marker) present the second best paths of the two latest survivor contentions (at time kT and $(k+1)T$). The black lines show the branches of the other non-survivor paths (which are not fully shown here as these paths are not involved in the soft information generation). It should be noted that the trellis memory, v, and the separation length between two best paths, δ_m, are 2 and 4 respectively in this example.

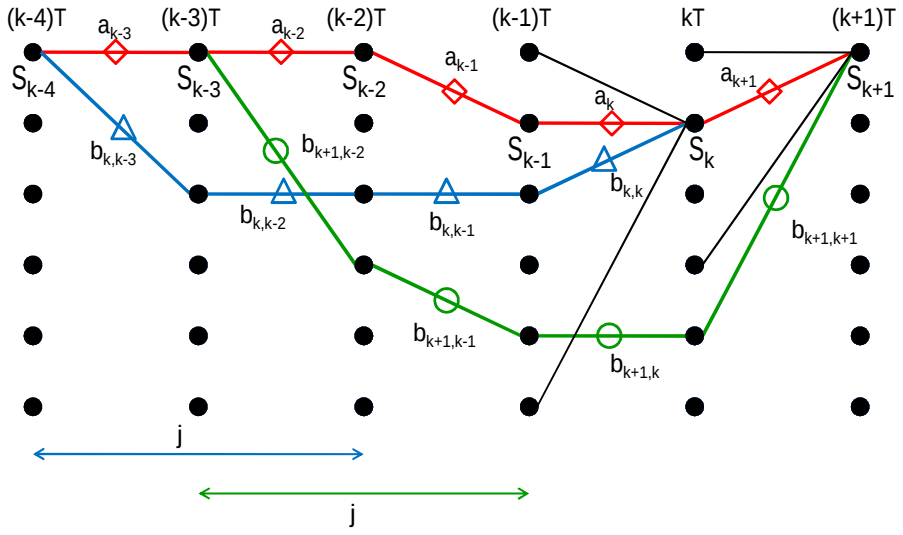


Figure 4.3: Example of the SOVA-based algorithm, where a_j is the survivor symbol, and $b_{i,j}$ is the symbol corresponding to the transition between time $(j-1)T$ and jT on the second best path (at state S_i when $t = iT$).

The algorithm starts by initializing the probability of making a wrong decision for each symbol along the survivor path to zero, $\hat{p}_j = 0$, where j is the symbol index. Then for each survivor state S_k at time kT:

1. Compute the path metric difference (Δ) between the survivor path and the minimum metric among the non-survivor paths. From this, the probability of selecting a wrong path, p_{sk}, at state S_k is approximately

$$p_{sk} = \frac{1}{1 + e^{\Delta}}. \tag{4.7}$$

2. For $j = k - v$ to $j = k - \delta_m$ (for the particular trellis shown in Fig. 4.3, $v = 2$ and $\delta_m = 4$):

   - If $a_j \neq b_{k,j}$ then update

   $$\hat{p}_j = \hat{p}_j(1 - p_{sk}) + (1 - \hat{p}_j)p_{sk}, \tag{4.8}$$

   It should be noted that $\hat{p}_j$ can be updated in each symbol period. For example, consider the updating process for two consecutive symbol time index, kT and $(k + 1)T$; we can see that the probability $\hat{p}_{k-2}$ is updated twice if a_{k-2} is different to both $b_{k,k-2}$ and $b_{k+1,k-2}$. It now becomes difficult to decide if $\hat{p}_{k-2}$ corresponds to $b_{k,k-2}$ or $b_{k+1,k-2}$, and which one of these should be stored in memory for later use (the bit reliability computation). Here, it is assumed that the probability corresponds to $b_{k+1,k-2}$ since it causes the latest update in $\hat{p}_{k-2}$. As a result of that, $b_{k+1,k-2}$ is stored in memory.

3. If $v > \delta_m$, then leave $\hat{p}_j$ as 0, and insert a copy of a_j into the memory slot where a b_j symbol should be if $v < \delta_m$.

At the end of the above process, for each survivor symbol a_j, there is a probability $\hat{p}_j$ that provides a degree of reliability to the symbol. However, the calculated symbol reliability can not be directly used by the convolutional decoder. This happens because such reliability are only meaningful for the symbols on the two best paths and does not associate with the symbols on the other two paths. In other word, the reliability information only takes into account half the number of symbols within the modulation constellation. Therefore, the convolutional decoder can not effectively operate on the computed soft information. A solution for this is to convert the symbol soft information to bit soft information. Given that there can only be two possible values for a bit, the soft information computed from two different symbols should have enough data to specify the reliability of the bit.

4.2.2 Converting Symbol Reliability to Bit Reliability

The conversion from symbol reliability to bit reliability is based on the standard conversion suggested in [67]. Each symbol a_k in the survivor path can be converted to a pair of bits, d_1d_2. According to [67] if we know the probabilities of all the symbols in the constellation, $P(a_k = 0)$, $P(a_k = 1)$, $P(a_k = 2)$, and $P(a_k = 3)$, then

the reliability of the first bit d_1 can be calculated as

$$\hat{L}(d_1) = \ln\left(\frac{\sum_{m=1d_2} P(a_k = m)}{\sum_{n=0d_2} P(a_k = n)}\right), \tag{4.9}$$

and that for the second bit as

$$\hat{L}(d_2) = \ln\left(\frac{\sum_{m=d_11} P(a_k = m)}{\sum_{n=d_10} P(a_k = n)}\right), \tag{4.10}$$

where m and n are symbols within the constellation.

It is not possible for the above process to be directly applied here because it requires all four symbol probabilities while there are only two probabilities available from the previous operation: $P(a_j|a_j, b_j) = 1 - p_j$, and $P(b_j|a_j, b_j) = p_j$ (where b_j is the stored symbol obtained from the second paths). Therefore, given that the symbol to bit mapping process is $0 = 00$, $1 = 01$, $2 = 10$ and $3 = 11$, the conversion process is modified to: For each hard decision a_j, we compare its bits with those of b_j,

- if the i^{th} bit of a_j and b_j are both 0, then the reliability of this i^{th} bit is set to $\hat{L}(d_i) = -E(|\hat{\boldsymbol{L}}|)$, where $\boldsymbol{L}$ is the bit reliability sequence,
- if the i^{th} bit of a_j and b_j are both 1, then the reliability of this i^{th} bit is set to $\hat{L}(d_i) = E(|\hat{\boldsymbol{L}}|)$,
- if the i^{th} bit of a_j is 1 while that of b_j is 0, then the reliability becomes

$$\hat{L}(d_i) = \ln\left(\frac{1 - p_j}{p_j}\right), \tag{4.11}$$

- Lastly, if the i^{th} bit of a_j is 0 while that of b_j is 1, then the reliability is now

$$\hat{L}(d_i) = -\ln\left(\frac{1 - p_j}{p_j}\right). \tag{4.12}$$

Once the conversion finishes, the bit reliability is used as a soft bit for the VA of the convolutional decoder. Note: the computation for the above algorithms can be reduced by replacing the value of $E(|\hat{\boldsymbol{L}}|)$ with a constant C/γ, where $C = max(|\hat{\boldsymbol{L}}|)$ and γ is a scale factor. Although this may slightly reduce the computation complexity, the performance can be degraded significantly if the scale factor γ is not

carefully chosen. Through various simulations, it was found that a value of γ near 2 provides better performance than those of other factors for the system considered in this thesis.

4.3 BCJR-based Algorithm

The classical BCJR algorithm [37] is based on a symbol-wise maximum a posteriori probability criteria and proves to be optimal for estimating the states or outputs of a Markov chain observed in white noise. Here, a BCJR-based algorithm derived especially for the predictive demodulator design is present. In a similar manner to the algorithms in [39, 43], the derivation of the BCJR-based algorithm starts with similar steps to those of the original algorithm. However, it differs from the classical one in the sense that in later derivation steps, it takes into account the trellis structure and design approximations of the predictive receiver. The algorithm has some similarities to those in [42, 43] as they all operate on reduced-size trellises in which a reliability calculation involves a past symbol sequence. While the algorithms in [42, 43] have to rely on the forward or backward updating process to identify the past symbol sequence, it is available to the BCJR-based algorithm from the PSP process.

Let $\boldsymbol{y}$ be the received sequence when $\boldsymbol{x}$ is transmitted over a wireless channel. In order to estimate the original message $\boldsymbol{a}$, like other MAP algorithms, the BCJR-based algorithm computes the a posteriori probability $P(a_n = k|\boldsymbol{y})$ for $n = 0, \ldots, N-1$, where k is the value of a symbol within the modulation constellation. This estimation is done with the help of a trellis. Each transition between state s' at time $(n-1)T$ and state s at the current time nT is due to a hypothesized symbol, a_n. Hence, the probabilities that $a_n = k$ given $\boldsymbol{y}$, are equal to the sum of probabilities of the transitions $\langle s', s\rangle$ corresponding to symbol k, hence,

$$\begin{aligned} P(a_n = k|\boldsymbol{y}) &= \sum_{R_k} P(\langle s', s\rangle|\boldsymbol{y}) \\ &= \sum_{R_k} P(\langle s', s\rangle, \boldsymbol{y})/P(\boldsymbol{y}). \end{aligned} \tag{4.13}$$

The notation R_k means the summation in (4.13) is computed over all the state transitions $\langle s', s\rangle$ that correspond to the message symbol $a_n = k$. Since the factor $P(\boldsymbol{y})$ is the same for all possible values of k in (4.13), it has no effect on the reliability

calculation and thus, can be left out. So

$$P(a_n = k|\boldsymbol{y}) = \sum_{R_k} P(\langle s', s\rangle, \boldsymbol{y}). \tag{4.14}$$

This probability is computed for all values of k.

The joint probability $P(\langle s', s\rangle, \boldsymbol{y})$ can be further expressed as

$$P(\langle s', s\rangle, \boldsymbol{y}) = P(\langle s', s\rangle, \boldsymbol{y}_0^{n-1}, y_n, \boldsymbol{y}_{n+1}^N). \tag{4.15}$$

According to the statistical relation, $P(A, B|C) = P(A|B, C)P(B|C)$,

$$\begin{aligned} P(\langle s', s\rangle, \boldsymbol{y}) &= P(\boldsymbol{y}_{n+1}^N|\langle s', s\rangle, \boldsymbol{y}_0^{n-1}, y_n)P(\langle s', s\rangle, \boldsymbol{y}_0^{n-1}, y_n) \\ &= P(\boldsymbol{y}_{n+1}^N|\langle s', s\rangle, \boldsymbol{y}_0^{n-1}, y_n)P(y_n, s|s', \boldsymbol{y}_0^{n-1})P(s', \boldsymbol{y}_0^{n-1}). \end{aligned} \tag{4.16}$$

Recall in the previous chapter we assumed that the relation between any two samples that are K samples apart is insignificant to the prediction process. In addition, the sequence $\boldsymbol{y}_{n+1}^N$ is independent of the previous state s' given the current state s is known. Thus, the three probabilities in (4.16) can be reduced to

$$P(\langle s', s\rangle, \boldsymbol{y}) = P(\boldsymbol{y}_{n+1}^N|, s, \boldsymbol{y}_{n-K+1}^n)P(y_n, s|s', \boldsymbol{y}_{n-K}^{n-1})P(s', \boldsymbol{y}_0^{n-1}). \tag{4.17}$$

If we assume that $N_{ss} = 1$, for this particular demodulation trellis, the probabilities in (4.17) can be defined as:

$$\alpha_{n-1}(s') = P(s', \boldsymbol{y}_0^{n-1}) \tag{4.18}$$

$$\gamma_n(\langle s', s\rangle) = P(y_n, s|s', \boldsymbol{y}_{n-K}^{n-1}) \tag{4.19}$$

$$\beta_n(s) = P(\boldsymbol{y}_{n+1}^N|s, \boldsymbol{y}_{n-K+1}^n) \tag{4.20}$$

At time nT, the probabilities α, γ, and β are associated with the past, the present and the future of the received sequence $\boldsymbol{y}$. So,

$$P(\langle s', s\rangle, \boldsymbol{y}) = \alpha_{n-1}(s')\gamma_k(\langle s', s\rangle)\beta_k(s). \tag{4.21}$$

Finally, in order to guarantee all the transition probabilities over all the trellis branches always sum to 1 at each time step nT, the joint probability must be normalized as

$$P_{norm}(\langle s', s\rangle, \boldsymbol{y}) = \frac{P(\langle s', s\rangle, \boldsymbol{y})}{\sum_k \sum_{R_k} P(\langle s', s\rangle, \boldsymbol{y})}. \tag{4.22}$$

Note: the notation γ is redefined in this section. It is now a probability instead of a scaling factor as its first definition in Section 4.2.2.

4.3.1 Calculation of γ

The algorithm starts with the computation for γ. According to the last section, $\gamma_n(\langle s', s\rangle) = P(y_n, s|s', \boldsymbol{y}_{n-K}^{n-1})$ is the probability that the received symbol is y_n at time nT and the current state is s, given the previous state was s', and the past sample sequence was $\boldsymbol{y}_{n-K}^{n-1}$. By applying the statistical formula $P(A, B|C) = P(A|B, C)P(B|C)$ again, $\gamma_n(\langle s', s\rangle)$ can be expressed as

$$\begin{aligned}\gamma_n(\langle s', s\rangle) &= P(y_n, s|s', \boldsymbol{y}_{n-K}^{n-1}) \\ &= P(y_n|\langle s', s\rangle, \boldsymbol{y}_{n-K}^{n-1})P(s|s', \boldsymbol{y}_{n-K}^{n-1}). \end{aligned} \tag{4.23}$$

Let us consider the meaning of each of these probability factors for this particular application. We have

$$\begin{aligned}P(s|s', \boldsymbol{y}_{n-K}^{n-1}) &= \frac{P(s, s'|\boldsymbol{y}_{n-K}^{n-1})}{P(s'|\boldsymbol{y}_{n-K}^{n-1})} = \frac{P(s, s'|\boldsymbol{y}_{n-K}^{n-1})P(\boldsymbol{y}_{n-K}^{n-1})}{P(s'|\boldsymbol{y}_{n-K}^{n-1})P(\boldsymbol{y}_{n-K}^{n-1})} \\ &= \frac{P(s, s')}{P(s')} = P(s|s'). \end{aligned} \tag{4.24}$$

Hence, $P(s|s', \boldsymbol{y}_{n-K}^{n-1}) = P(a_{n-1})$.

As far as the other factor, $P(y_n|\langle s', s\rangle, \boldsymbol{y}_{n-K}^{n-1})$ is concerned, the joint occurrence of the consecutive states s' and s is equivalent to the specification of a PEF filter, which corresponds to the symbol sequence, $\boldsymbol{a}_{n-L-K+1}^{n-1}$. This means $P(y_n|\langle s', s\rangle, \boldsymbol{y}_{n-K}^{n-1})$ is equivalent to $P(y_n|\boldsymbol{y}_{n-K}^{n-1}, \boldsymbol{a}_{n-L-K+1}^{n-1})$. Thus,

$$\gamma_n(\langle s', s\rangle) = P(y_n|\boldsymbol{y}_{n-K}^{n-1}, \boldsymbol{a}_{n-L-K+1}^{n-1})P(a_{n-1}). \tag{4.25}$$

From (3.6) and the assumption that each symbol in the modulation has equal probability of transmission, the calculation of $\gamma_n(\langle s', s\rangle)$ simplifies to

$$\gamma_n(\langle s', s\rangle) = \frac{1}{4}\exp\Big(-\mu_n\Big(\boldsymbol{a}_{n-L-K+1}^{n-1}\Big)\Big). \tag{4.26}$$

With the application of PSP, the computation of $\mu_n\Big(\boldsymbol{a}_{n-L-K+1}^{n-1}\Big)$ can only be done for the most likely sequence $\boldsymbol{a}_{n-L-K+1}^{n-1}$. Therefore, the computation for $\gamma_n(\langle s', s\rangle)$ is limited to such a sequence.

4.3.2 Calculation of α

By definition,

$$\alpha_n(s) = P(s, \boldsymbol{y}_0^n) = P(s, \boldsymbol{y}_0^{n-1}, y_n). \tag{4.27}$$

By applying the probability identity, $P(A) = \sum_B P(A, B)$, where the summation includes all values of B, we have

$$\begin{aligned} \alpha_n(s) &= P(s, \boldsymbol{y}_0^{n-1}, y_n) \\ &= \sum_{s'} P(\langle s', s\rangle, \boldsymbol{y}_0^{n-1}, y_n) \\ &= \sum_{s'} P(s, y_n | s', \boldsymbol{y}_0^{n-1}) P(s', \boldsymbol{y}_0^{n-1}). \end{aligned} \tag{4.28}$$

If the previous state (and its PSP symbol sequence) is given, the current state and current symbol do not depend on the entire sequence $\boldsymbol{y}_0^{n-1}$. They instead rely only on the much shorter sequence, $\boldsymbol{y}_{n-K}^{n-1}$ required for the prediction process. This is because it is assumed that any relation between samples that are more than KT apart is negligible. Therefore,

$$\begin{aligned} \alpha_n(s) &= \sum_{s'} P(s, y_n | s', \boldsymbol{y}_{n-K}^{n-1}) P(s', \boldsymbol{y}_0^{n-1}) \\ &= \sum_{s'} \gamma_n(\langle s', s\rangle) \alpha_{n-1}(s'), \end{aligned} \tag{4.29}$$

where $\gamma_n(\langle s', s\rangle) = P(s, y_n | s', \boldsymbol{y}_{n-K}^{n-1})$. With the application of PSP, there is only one survivor path for each previous state s'. This means that the specification of s' alone is enough for retrieving the most likely symbol sequence $\boldsymbol{a}_{n-L-K+1}^{n-2}$. An example of the computation process of $\alpha_n(s)$ for a demodulation trellis configured with $K = 3$, $L = 3$, and $P = 2$ is shown in Fig. 4.4.

In order to avoid overflow or underflow it is very important that $\alpha_n(s)$ is normalized. This can be done by dividing $\alpha_n(s)$ by its sum over all state transitions, that is

$$\alpha_n(s) = \frac{\alpha_n(s)}{\sum_s \alpha_n(s)}. \tag{4.30}$$

This normalization process does not change the final soft output result since all $\alpha_n(s)$ are scaled by the same amount. The initial conditions for $\alpha_0(s)$ are

$$\alpha_0(s) = \begin{cases} 1 & \text{if } s = 0, \\ 0 & \text{if } s \neq 0. \end{cases}$$

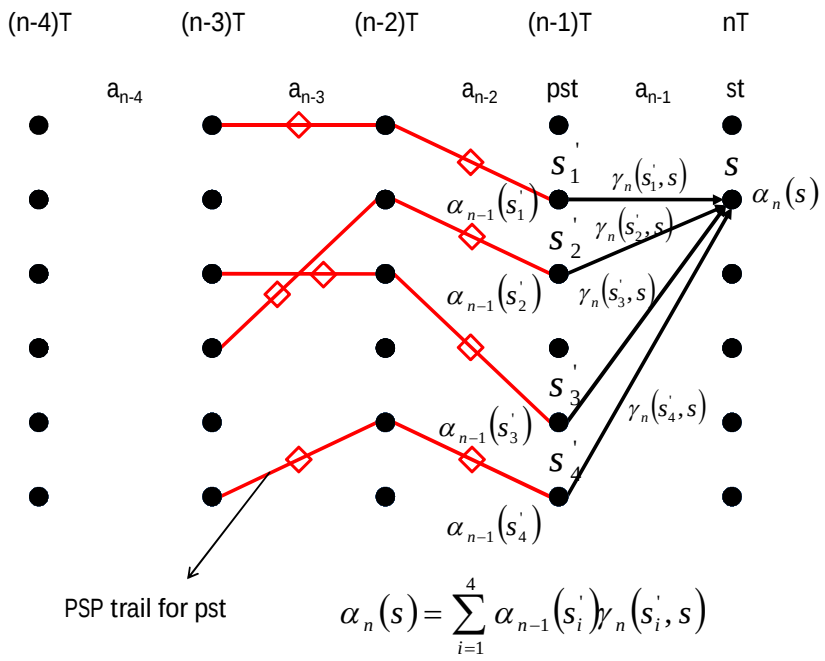


Figure 4.4: Trellis-aided recursive calculation of α.

These initial conditions correspond to a terminated trellis, that is, it begins and ends in the 0 state. We can modify the transmitted data sequence to make this happen. This may slightly alter the generality of the transmitted sequence.

4.3.3 Calculation of β

The recursive formulation of β is obtained in the same way as α. By definition $\beta_n(s) = P(\boldsymbol{y}_{n+1}^N | s, \boldsymbol{y}_{n-K+1}^n)$, which can be re-written as $\beta_{n-1}(s') = P(\boldsymbol{y}_n^N | s', \boldsymbol{y}_{n-K}^{n-1})$. By applying the probability identity to the RHS of the formula, we obtain

$$\begin{aligned}
\beta_{n-1}(s') &= \sum_s P(s, y_n, \boldsymbol{y}_{n+1}^N | s', \boldsymbol{y}_{n-K}^{n-1}) \\
&= \sum_s P(\boldsymbol{y}_{n+1}^N | \langle s', s \rangle, y_n, \boldsymbol{y}_{n-K}^{n-1}) P(s, y_n | s', \boldsymbol{y}_{n-K}^{n-1}) \\
&= \sum_s P(\boldsymbol{y}_{n+1}^N | \langle s', s \rangle, \boldsymbol{y}_{n-K+1}^n, y_{n-K}) P(s, y_n | s', \boldsymbol{y}_{n-K}^{n-1}) \qquad (4.31)
\end{aligned}$$

Since any received data sample after time nT is not influenced by those samples that are K sample times before it, as discussed above, the sample sequence $\boldsymbol{y}_{n+1}^{N}$ is not dependent on y_{n-K}. On the other hand, if the current state is given, the sequence does not depend on the previous state s'. Hence,

$$\begin{aligned}\beta_{n-1}(s') &= \sum_{s} P(\boldsymbol{y}_{n+1}^{N}|s, \boldsymbol{y}_{n-K+1}^{n}) P(s, y_n|s', \boldsymbol{y}_{n-K}^{n-1}) \\ &= \sum_{s} \beta_n(s)\gamma_n(\langle s', s\rangle) \end{aligned} \tag{4.32}$$

Figure. 4.5 illustrates this updating process for a demodulation trellis with $K = 3$, $L = 3$, and $P = 2$.

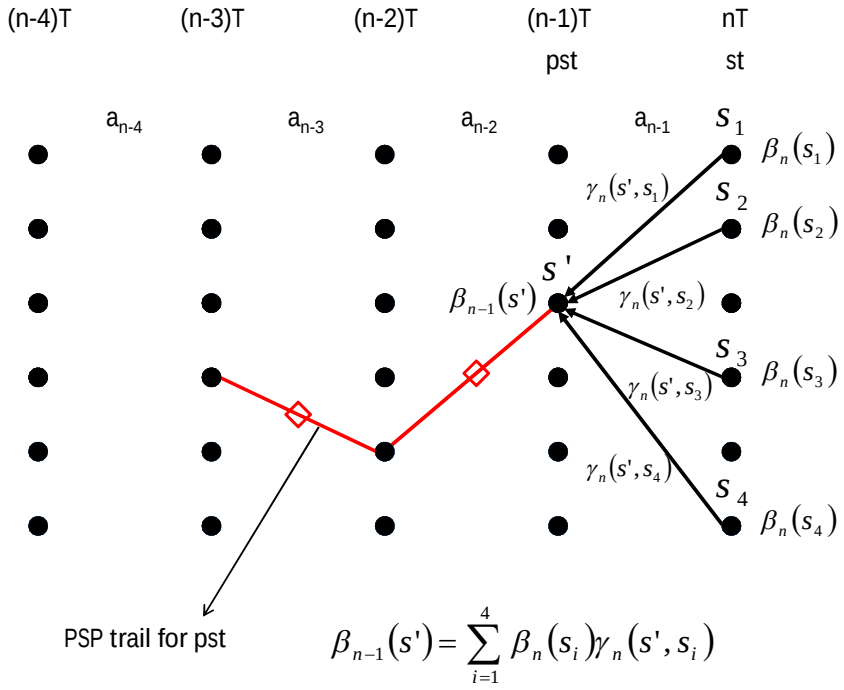


Figure 4.5: Trellis-aided recursive calculation of β.

The process of normalization is also applied to β in a similar way to that of α, giving

$$\beta_{n-1}(s') = \frac{\beta_{n-1}(s')}{\sum_{s'} \beta_{n-1}(s')}. \tag{4.33}$$

From the moment we know $\beta_N(s)$, we can compute the other values of β. The initial conditions for this trellis, which is all-zero terminated, are

$$\beta_N(s) = \begin{cases} 1 & \text{if } s = 0, \\ 0 & \text{if } s \neq 0. \end{cases}$$

4.3.4 Delivering BCJR Soft Information to A Decoder

At the moment, the BCJR algorithm has provided the receiver with the probability $P(a_n = k|\boldsymbol{y})$ for all values of k at each time step in the demodulation matrix. This information is useful, but is hard to use directly in the decoder. What can be done here is to convert the information into symbol or bit log-likelihood ratios that the decoder can understand. Having said that, I have chosen the bit log-likelihood ratios as the input to the decoder instead of that for symbols. This is because the decoder has already been configured to work with bit log-likelihood ratio. In addition, the SOVA-based algorithm developed earlier used bit log-likelihood ratios. It will be easier to compare BCJR and SOVA performances if their outputs are in the same form.

Because each symbol corresponds to a dibit in the CPM modulation, the log-likelihood ratio is computed for each bit in the dibit from the four probabilities $P(a_n = k|\boldsymbol{y})$ available in each time step. Let us denote $k = d_1 d_2$, where $d_1 d_2$ is a pair of bits corresponding to the symbol value k at time nT. The bit log-likelihood ratio $L(d_i)(n)$, $i = 1, 2$, is then calculated as

$$L(d_i)(n) = \frac{\sum_{d_i=1} P(a_n = k|\boldsymbol{y})}{\sum_{d_i=0} P(a_n = k|\boldsymbol{y})}, \tag{4.34}$$

where the summation in the numerator is computed over all the probabilities correspond to a k value whose i^{th} bit is 1 and the i^{th} bit is 0 in the case of the summation in the denominator.

4.4 Iterative Detection

The iterative system developed in this thesis is shown in Fig. 4.6.

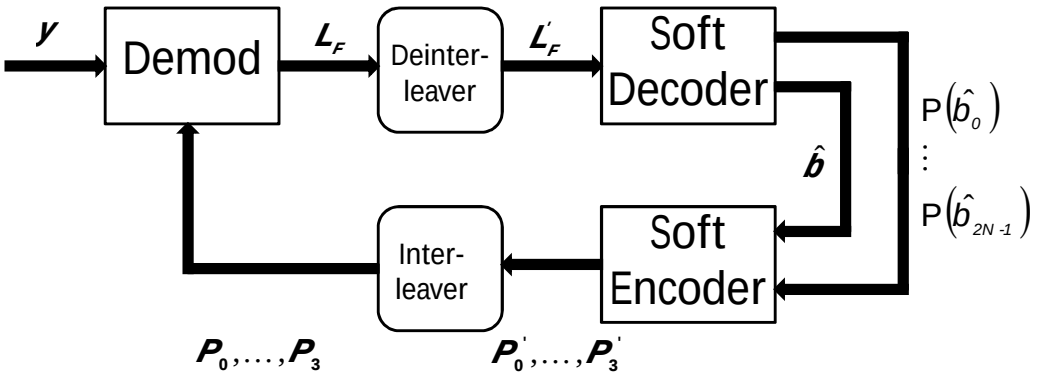


Figure 4.6: Iterative system for predictive detection.

In general, it is quite similar to other iterative systems found in [89, 90] in the sense that the demodulator calculates reliability information in the form of bit log-likelihood ratios ($\boldsymbol{L}_F$), which are deinterleaved ($\boldsymbol{L}'_F$) before passing them on to the soft decoder. The decoder is able to generate reliability information ($P(\hat{b_n})$) as well as hard decisions ($\hat{\boldsymbol{b}}$). From this point onward, the system differentiates itself from the ones in [89, 90] since $P(\hat{b_n})$ cannot be fed directly (through the interleaver) to the demodulator. This is mainly because the demodulator works on encoded symbols while the decoder outputs uncoded data bits. Hence, in order for this feedback process to happen, the soft encoder is present in the feedback path.

The encoder is based on the structure of the ordinary convolutional encoder used in the system. Its task is to use the inputs of the hard decision bits and their corresponding probabilities to compute the probabilities for each of the possible values of a transmitted symbol at a given time. So, at the output of the soft encoder, there are 4 probability sequences $\boldsymbol{P}'_0$,..., $\boldsymbol{P}'_3$ whose n^{th} elements, $P_0^n, ..., P_3^n$, are the probabilities that the n^{th} transmitted symbol is 0, 1, 2 or 3. These sequences are interleaved (to become $\boldsymbol{P}_0$,..., $\boldsymbol{P}_3$) before being fed back into the demodulation process.

In this section, the process of using the soft encoder to compute encoded-symbol probability will be described first, and then I will show how the hard and soft encoded information is used to enhance the performance of the demodulator.

4.4.1 Feedback Soft Information Calculation

By applying a reliability generation algorithm in the decoder, thus effectively turning it into a soft output decoder, it is able to deliver both hard decision bits $\hat{\boldsymbol{b}}$ and their corresponding probabilities of being right $P(\hat{b_n})$ for the un-coded data. As discussed earlier, these outputs are then delivered to the soft encoder to compute the probability P_i^n for each of the possible transmitted (encoded) symbols. This calculation is relatively complex depending on the convolutional code used in the system, especially its trellis. Here, I will show such a process for the rate 1/2 code mentioned earlier. In comparison, the code has a smaller trellis than that of its rate three quarter counterpart so its computation for P_i^n is considerably simpler.

It is known that each state in the trellis of the rate 1/2 code is a symbol (or dibit) and the input to the code in each time period is also a symbol. So a branch in such a trellis can be specified by a set of 2 symbols (or 2 dibits). In addition, the symbol corresponding to the next state is also the current input symbol. Hence, each transition in the trellis is specified by a sequence of 2 data symbols (or 4 data bits). Since the probability of being right for each un-coded data bit $P(\hat{b_n})$ is known, it is possible, at every time step, to compute the probabilities for all possible branches instead of considering a single branch like a normal hard encoding process. Because each transition outputs a distinct pair of symbols, the probability of the symbol pair is that of the transition. This means that for each time step the soft encoder can acquire the probabilities for all possible symbol pairs. In addition, scaling is applied so that, at each time period, the sum of all the probabilities equals 1. At the end of the whole process, the soft encoder outputs 4 sequences of probabilities ($\boldsymbol{P}'_0$,..., $\boldsymbol{P}'_3$) that contain the transmission probabilities for each of the hypothesized symbols for a transmitted frame.

The following example shows how the process works for the first time step with an input data bits sequence of [1 0 0 0] and its corresponding sequence of probabilities, [0.3 0.8 0.5 0.5]. The bit sequence corresponds to a symbol pair, [2 0]. Given that the encoding starts with an initial state of 0, for such symbol inputs, the encoding path for the standard encoder is given in Fig. 4.7. In this case, all the possible transitions are considered to determine their probabilities. For the very first time step, as state 0 is a common state, there are only 4 transitions to consider.

As discussed earlier, each transition is specified by a sequence of 4 bits in which the first two bits represent the current state and the others represent the current symbol input. Normally, the probability of each transition is calculated based on the probabilities of all the bits. Because of the common state, the first two bits of any transition in the first time step are fixed as [0 0], therefore these transition probabilities are determined by the two remaining bits. Table. 4.1 displays all possible transition-represented bit sequences, the corresponding output symbol (bit) sequence and probabilities for the first time step.

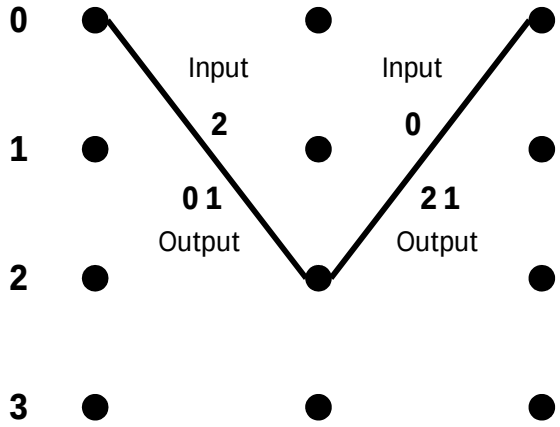


Figure 4.7: Encoding path for data bit sequence [1 0 0 0].

Table 4.1: Example of first transition bit sequences, outputs, and probabilities.

Transition in Bit Sequence (State, Input Symbol)	Output in Bits (Symbols)	Transition Probability
0 0, 0 0 (0, 0)	0 0 1 0 (0 2)	(1 - 0.3) × 0.8 = 0.56
0 0, 0 1 (0, 1)	1 1 0 0 (3 0)	(1 - 0.3) × (1 - 0.8) = 0.14
0 0, 1 0 (0, 2)	0 0 0 1 (0 1)	0.3 × 0.8 = 0.24
0 0, 1 1 (0, 3)	1 1 1 1 (3 3)	0.3 × (1 - 0.8) = 0.06

Having computed all the possible output symbol pairs and their probabilities for the first time step, we can now calculate the overall probability for each possible symbol in the output. Because the first pair and the actual (third) output pair have the same first symbol of 0, the probability of the first output symbol being 0 is $P(\hat{a}_0 = 0) = 0.56 + 0.24 = 0.8$. Similarly, the second and fourth output pairs have a common first symbol of 3, thus, $P(\hat{a}_0 = 0) = 0.14 + 0.06 = 0.2$. Since symbols 1 and 2 do not appear in the first slot of any output pairs, $P(\hat{a}_0 = 1) = P(\hat{a}_0 = 2) = 0$. Table. 4.2 shows the probabilities for each possible output symbol value in the first time step.

Table 4.2: Example of symbol values probabilities for first time step.

Possible symbol values	Probability $P(\hat{a}_0 = k)$	Probability $P(\hat{a}_1 = k)$
k = 0	0.56 + 0.24 = 0.8	0.14
k = 1	0	0.24
k = 2	0	0.56
k = 3	0.14 + 0.06 = 0.2	0.06

The process of computing these probabilities for the next (and later) time steps is very similar to that of the first time step. The only difference is in the number of transitions that we need to consider. There are 16 transitions (with 4 common states) in the main body of the trellis. This means that the computational complexity of this and the following time steps is approximately 4 times that of the first one. It should be noted here that although normalization was not applied for the first time step (as the values naturally add up to 1), such a process may need to be done at the end of the process for later time steps.

It should be noted that the subtraction of intrinsic information is not included in the feedback loop because the intrinsic information received by the demodulator is in the form of symbol probabilities (as shown in the following section) while the demodulator output is in the form of bit log likelihoods. On the other hand,

computing the extrinsic information from the decoding part can be done but does not yields better performance as the soft information saturates after the second iteration. Although, this is not often the case for iterative detection, similar systems do exist in literature. A notable example of this is the receiver design in [36].

4.4.2 Demodulation Process with Feedback Information

In general, the predictive demodulation process can be divided into 2 parts:

- Calculating the path metrics entering each state in the trellis.
- Searching for the best path and computing soft information (or hard decisions) based on the search result.

Because the original design of the path metric derivation given in (3.17 and 3.18) does not include soft information, it is necessary to modify it for iterative detection.

The modification was done by trying to optimise the probability $P(\boldsymbol{y}, \boldsymbol{a})$ instead of $P(\boldsymbol{y}|\boldsymbol{a})$, then (3.2) becomes

$$P(\boldsymbol{y}, \boldsymbol{a}) = P(y_{M-1}|\boldsymbol{y}_0^{M-2}, \boldsymbol{a}) \dots P(y_0|\boldsymbol{a}) P(a_{N-1}) \dots P(a_0). \tag{4.35}$$

Then the log-likelihood metric in (3.4) changes to

$$\ln(p(\boldsymbol{y}, \boldsymbol{a})) = \sum_{n=0}^{N-1} \left(\ln(P(a_n)) + \sum_{j=0}^{N_{ss}-1} \ln(p(y_{nN_{ss}+j}|\boldsymbol{y}_0^{nN_{ss}+j-1}, \boldsymbol{a})) \right). \tag{4.36}$$

Hence, the branch metric is now

$$\mu_n\left(\boldsymbol{a}_{n-L-K+1}^{n-1}\right) = \ln\left(\sigma_n^2(\boldsymbol{a}_{n-L-K+1}^{n-1})\right) + \frac{\left|\tilde{\boldsymbol{w}}^\dagger(\boldsymbol{a}_{n-L-K+1}^{n-1})\boldsymbol{y}_{n-K}^{n}\right|^2}{\sigma_n^2(\boldsymbol{a}_{n-L-K+1}^{n-1})} - \ln(P(a_n)). \tag{4.37}$$

Because of the difference between the demodulation trellis and decoder trellis, and hence their corresponding reliability information being different, a compensating scaling factor c needs to apply to the soft factor $\ln(P(a_n))$ in (4.37). The actual demodulation branch metric in the iterative structure is then

$$\mu_n\left(\boldsymbol{a}_{n-L-K+1}^{n-1}\right) = \ln\left(\sigma_n^2(\boldsymbol{a}_{n-L-K+1}^{n-1})\right) + \frac{\left|\tilde{\boldsymbol{w}}^\dagger(\boldsymbol{a}_{n-L-K+1}^{n-1})\boldsymbol{y}_{n-K}^{n}\right|^2}{\sigma_n^2(\boldsymbol{a}_{n-L-K+1}^{n-1})} - c\ln(P(a_n)). \tag{4.38}$$

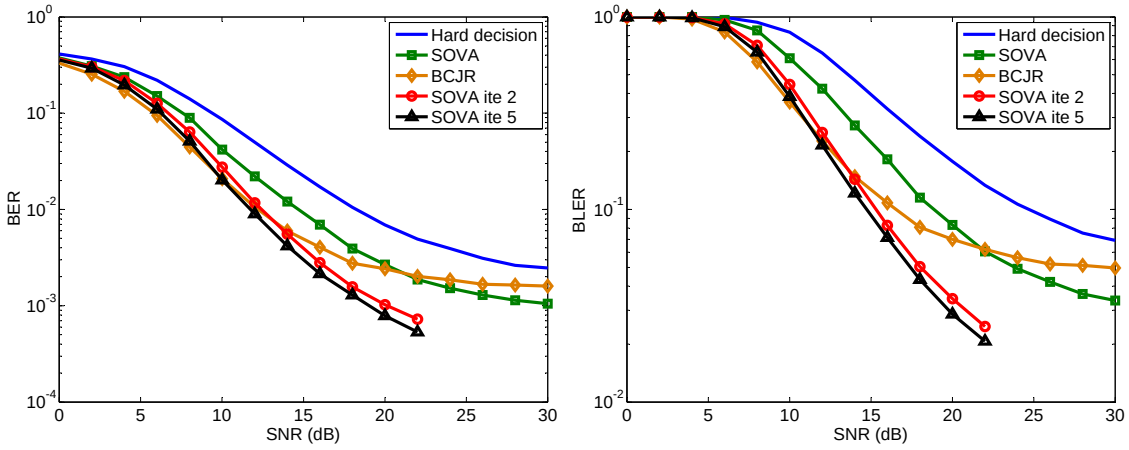
4.5 Simulation Results

The performance of the predictive system with soft and iterative detection is simulated with a coefficient LUT configured for a prediction order of $K = 3$, C4FM modulation with $L = 3$ and two equal path Rayleigh channel that has a maximum Doppler shift $f_{d_{max}} = 50\ Hz$ $(0.014T)$, delay spread $\tau = 12.5\ \mu s$ and a SNR $\Upsilon = 30$ dB. This setting is chosen because it was shown in [14] that such a configuration can deliver one of the best overall performances across a variety of channel conditions and SNR. The receiver is configured to have a 16 state demodulation trellis; in other word, the PSP parameter $P = 2$. As discussed in Chapter 2, the channel model used for simulation is a Rayleigh channel with 2 equal power paths. Simulations are done with the transmission of baseband C4FM signal (whose characteristics were described in Chapter 2) over the Rayleigh channel. Results obtained are in the form of BER and BLER for different SNR, delay spread and maximum Doppler shift of the channel model. Encoded data is transmitted in blocks of 98 symbols.

4.5.1 Effect of SNR

For determining the effect of SNR on the system performance when the rate 1/2 code is used, the maximum Doppler shift and delay spread are fixed at 50 Hz and 50 μs. This creates a fast fading channel with high delay spread (approximately a quarter of T), which is often the case for large cell multicast network for public safety operations. Without additional equalization, a standard receiver may not be able to establish communication (even for voice transmission) for such a channel. Therefore, this is good for testing the limit of the predictive receiver and its enhancement as well as investigating if good SNR can help the receiver to deliver accurate data detection.

Figure 4.8 compares the predictive receiver BER and BLER performance with hard, soft and SOVA-based iterative detection using the rate 1/2 code. It is clear from these plots that soft and iterative detection have provided significant gains over hard detection. In term of soft detection, both the SOVA-based and BCJR-based algorithms allow a reduction in SNR of more than 5 dB at BER near 10^{-3} (a more detailed summary on the improvement is shown in Table. 4.3). The SOVA-based algorithm achieves considerably better detection performance costing only a small amount of additional complexity over the hard detection design. In a comparison



(a) BER Result.

(b) BLER Result.

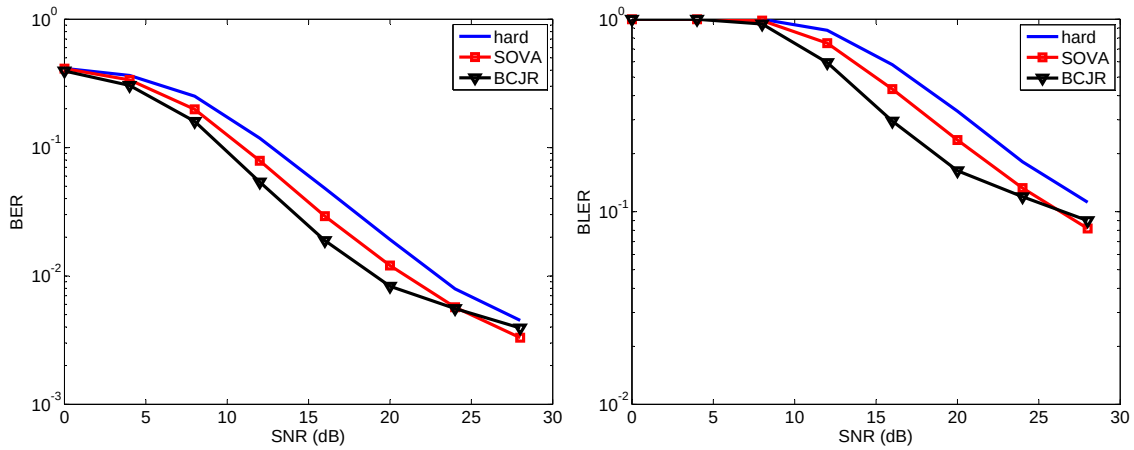
Figure 4.8: SNR results with rate 1/2 code, $\tau = 50\ \mu s$, and $f_{d_{max}} = 50\ Hz$.

between the two soft output algorithms, it can be seen that the BCJR-based algorithm performs better than the SOVA-based algorithm at SNRs below 20 dB and it is even comparable to those of iterative detection at lower SNR. This is expected since MAP-type algorithms are optimum for signal detection under Gaussian noise. However, Fig. 4.8 also shows that the SOVA-based algorithm has a better error floor at high SNRs than other algorithm. This is a very interesting result as MAP-type algorithms normally have lower error floors than those of SOVA-type algorithms. It will be shown later that the main reason behind this is the weaker tolerance to delay spread of the BCJR-based algorithm when compared to that of the SOVA-based algorithm for this particular receiver.

Although soft detection schemes do significantly enhance detection accuracy, it is still hard to establish communication for this particular channel condition since their BERs have not reached values below 10^{-3}. As can be seen from Fig. 4.8, both soft detection schemes reach their error floor around 30 dB so it is very hard for the BER curve to go down further by simply increasing SNR. One of the few effective methods to break through the error floor of soft detection is the application of closed-loop detection. As shown in Section 4.4, the application of closed-loop detection increases the computational demand of the overall system significantly, especially with a high number of iterations. Therefore, the closed-loop system is

constructed based on the simpler SOVA-based algorithm (applied to both the demodulator and decoder) so that the overall design complexity is not too high to be implementable on mobile devices.

The results obtained provide a clear indication that the system performance improves significantly with iterative detection. Its SNR gain is approximately twice that of soft detection with the SOVA-based algorithm and it is able to achieve BER $< 10^{-3}$ with SNR at 22 dB. Although the BER values obtained from closed-loop detection may not be low enough to be useful for data transmission, the system is now able to establish very reliable voice transmission. In comparison, the two curves corresponding to 2 and 5 iterations ('ite 2' and 'ite 5') do not differ much from each other. This means that most gain from this closed-loop structure can be achieved with 2 iterations. Hence, in practice, there is no need to increase the number of iteration beyond 2 as each iteration adds a considerable amount of complexity into the system. Note: 2 iterations mean 2 times through the receiver due to 1 time feeding back.



(a) BER Result.

(b) BLER Result.

Figure 4.9: SNR results with rate 3/4 code, $\tau = 20\ \mu s$, and $f_{d_{max}} = 50\ Hz$.

Figure 4.9 shows the performance of the soft output algorithms with the rate three quarter code. Since this code is considerably weaker than the half rate one, a less severe channel setting of $\tau = 20\ \mu s$ and $f_{d_{max}} = 50\ Hz$ is applied in its sim-

ulation. By comparing the error curves in Fig. 4.8 with those in Fig. 4.9, it can be seen that the rate three quarter code is much weaker than the rate 1/2 code as expected. This comes from the fact that the hard decision BER and BLER results for SNR $> 25\ dB$ are approximately the same even though the channel setting here is much better than that of Fig. 4.8.

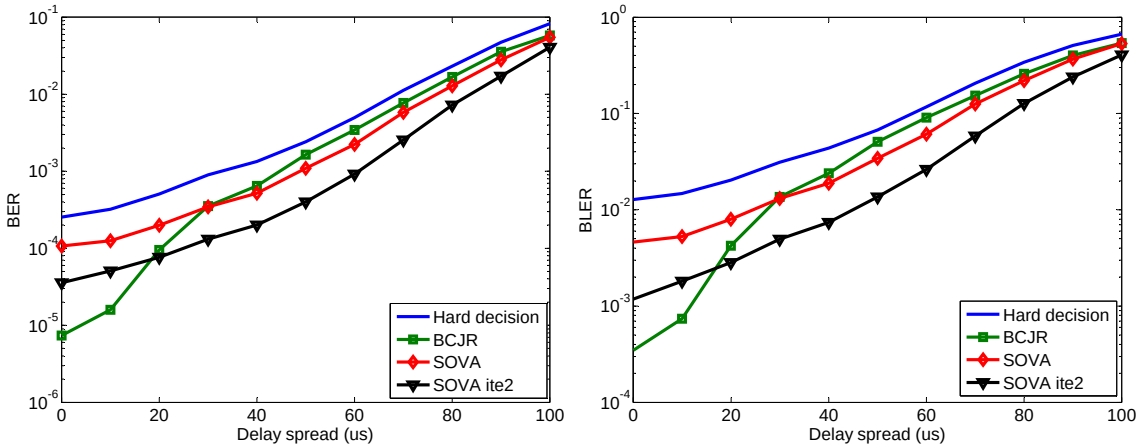
In addition, when examining Table. 4.3, which compares the SNR gains from using soft detection with both codes, we can see that the improvements corresponding to the rate three quarter code are considerably smaller than those corresponding to the other code. For example, when comparing their performances, at BER = 1%, the SOVA-based algorithm produces a gain of 2 dB over hard decision decoding instead of a gain of approximately 3.5 dB for the previous code. In the case of the BCJR-based algorithm, the gain is near 4 dB instead of being greater than 6 dB. Such a reduction can happen because being a weaker code, its decoder can not process the soft information as effectively as the other decoder. In other words, the system gain from using soft detection not only relies on the reliability generation algorithm, but also depends on the strength of the error control code. It should be noted that iterative detection is not considered here because the rate three quarter code is significantly more complex than the rate one half code, and may lead to a very demanding computational process for iterative detection.

Table 4.3: SNR gains over hard decision at different BERs.

	SOVA		BCJR		SOVA ite 2
	3/4 code	1/2 code	3/4 code	1/2 code	1/2 code
5% BER	2.02 dB	2.48 dB	3.55 dB	4.31 dB	3.41 dB
1% BER	1.96 dB	3.56 dB	3.85 dB	6.05 dB	5.87 dB
0.5% BER	2.61 dB	4.74 dB	2.31 dB	6.98 dB	7.59 dB

4.5.2 Effect of Delay Spread

The effect of channel delay spread on the reception process is also investigated by setting the Doppler effect to $f_{d_{max}} = 50\ Hz$ and SNR $\Upsilon = 30\ dB$. Such a high value of SNR is chosen so that the capability for handling delay spread of the system can be fully exploited under a fast fading scenario.



(a) BER Result.

(b) BLER Result.

Figure 4.10: Delay spread results with rate 1/2 code, $\Upsilon = 30\ dB$, and $f_{d_{max}} = 50\ Hz$.

Figure 4.10 illustrates the error performance of different detection schemes for a wide range of delay spread. It can be seen that apart from the BCJR curve, the other error curves have a steady (near-linear) increment with delay spread value. As expected, both soft detection methods show improvements over hard decision. However, these improvements are minor as the channel delay spread approaches 100 μs. This happens because such high delay spread value can cause quite severe inter-symbol interference, which can not be effectively overcome with soft detection. In comparison, the SOVA-based algorithm performs better than the BCJR-based algorithm for $\tau > 30\ \mu s$ but the latter significantly outperforms it (and even iterative detection) when the value of τ is low. This indicates that the BCJR-based algorithm is the best candidate and can provide outstanding performance for a channel whose delay spread is low, but it is clearly not a good choice for higher delay spread. On the other hand, the SOVA-based algorithm can work well over a wide range of delay spreads. Its performance is, however, quite steady for a wide range of delay spread and does not have a great improvement when moving from a multipath to a single-path channel. The results here also explain why in Fig. 4.8, the BCJR BER and BLER curves are inferior to the SOVA BER and BLER curves at high SNR where the effect of delay spread is the major source of detection errors. It should be noted here that the BCJR-based algorithm degradation at high delay spread is

not due to the error of the chosen LUT. It was found out that regardless of what LUT is used in the demodulation, the algorithm is still inferior to the SOVA-based algorithm at high delay spread.

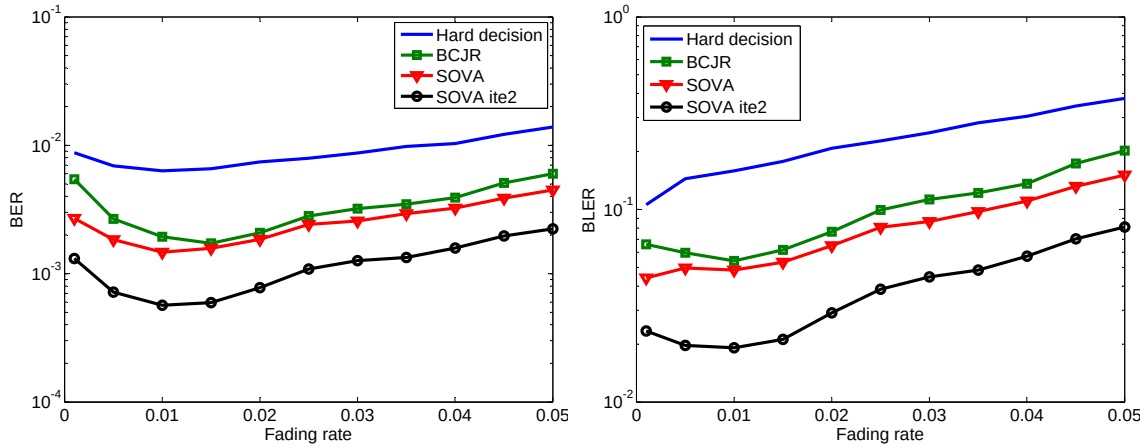
In terms of iterative detection, with 2 iterations, the gain in delay spread is approximately twice that of SOVA-based detection, which effectively provides a great boost to system performance, especially for mid to low delay values of τ. For example, hard decision can only achieve a BER value of 10^{-3} or lower when the value of τ does not exceed 30 μs. Iterative detection allows the system to achieve the BER with a much higher delay spread value of 60 μs. So, in this case, the system capability for handling delay spread is doubled and as a consequence, its operational range is enhanced significantly. This highlights the importance and effectiveness of iterative techniques in fast fading multipath channels. It is also noticeable that the iterative plots have very similar shapes to those of SOVA. This may happen due to the closed-loop structure based solely on the SOVA-based algorithm.

4.5.3 Effect of Doppler Frequency

Figure 4.11 shows the accuracies of different reception methods when the channel fading rate (as a result of Doppler shift), $f_{d_{max}}T$, increases from near static condition to very fast fading. The simulation is run with a relatively high delay spread of 50 us and a high SNR of 24 dB to reflect a large cell multicast network with high signal strength.

We can see that all the obtained BER and BLER curves do not vary much over a wide range of fading rate. This gives an indication that the original structure of the receiver is very effective against Doppler shift and the introductions of soft/iterative techniques help to shift the error rates down. In particular, iterative detection again shows a very prominent improvement by reducing the average BER from the order of 1% to 0.1%, while the application of soft detection achieves an average BER of approximately 0.4%.

An interesting feature that can be seen from all the error curves is that the best error rate of each curve is achieved near the fading rate of 0.01 instead of the lowest end like other receivers. This happens because the coefficient LUT is constructed



(a) BER Result.

(b) BLER Result.

Figure 4.11: Fading rate results with rate 1/2 code, $\tau = 50\ \mu s$, and $\Upsilon = 24\ dB$.

assuming a channel fading rate setting of $0.014T$. Hence, it is expected that the detection accuracy will be highest around such a value.

4.6 Summary

In this chapter, two reliability generation algorithms and an iterative structure for CPM predictive detection have been presented. A low complexity algorithm is developed based on the SOVA algorithm and complexity reduction is achieved by considering only the 2 best paths among all the ones competing at a particular state. This algorithm is shown to provide a considerable gain in performance and is quite suitable for high delay spread channels, which are often the case for multicast transmission or long range communication. On the other hand, the high complexity BCJR-based algorithm has been shown to outperform the other algorithm in mid to low SNR conditions or when the delay value is relatively low, $\tau < 0.15T$. Unfortunately, its performance degrades quickly with the increase of delay spread leading to its detection accuracy being less than that of the SOVA-based algorithm for higher delay spread condition. Although being quite different in structure and performance, both algorithms have shown clear improvements over the original hard decision decoding, especially with the rate 1/2 convolutional code. In addition, it has

been shown from simulation results that the performance gains for these algorithms are quite consistent over a wide range of high Doppler shift, which corresponds to very fast channel fading.

The developed iterative receiver structure has provided significant improvements in terms of fading rate, SNR and delay spread. In comparison to hard decision decoding, it is able to achieve a SNR reduction of more than 10 dB or a delay spread increment of more than 15 μs (which is more than 7% of the symbol period) at certain values of BERs for a multipath fast fading channel. The closed-loop system is currently constructed from the SOVA-based algorithm and has achieved most of its potential with only 2 iterations. The iterative technique can also be implemented with the BCJR-based algorithm and/or rate 3/4 code. However, subject to the estimation of resource requirements, I believe a SOVA-based iterative detection with the rate 1/2 code is a much more practical design for the P25 network. Therefore, it was the only one considered here.

Chapter 5

Diversity and Multiple LUTs Usage

As we have seen previously, a fast fading multipath channel with AWGN poses problems in the receiving and decoding of the CPM signal due to interference and the variability of the channel, which prevents accurate data estimation. It was seen in Chapter 4 that this can be reduced by using a reliability generation algorithm for constructing soft or iterative detection. Another suitable approach to improve the performance seen in Chapter 4 is diversity at the receiver. Diversity can be implemented in the time, frequency or spatial domains. Here we focus on the use of spacial diversity employing multiple antennas at the receiver. In addition, we introduce a new implementation for the predictive demodulator so that multiple coefficient look up tables (or LUTs) can support the demodulation process at the same time, and thus, enhance its performance. The first part of this chapter is focused on the design of a suitable diversity combining scheme that is not only low in complexity but is also able to deliver good performance. The other part of this chapter is dedicated to the development of the multiple LUTs usage method as an improvement on the demodulator design as well as a proof of the existing symmetry across the real axis within any coefficient LUT.

5.1 Diversity Combining Techniques Review

Consider a spatial diversity system with N receive antennas. The sample of a signal received by the j^{th} antenna may be written as

$$r_j = g_j s + n_j, \tag{5.1}$$

where s corresponds to the transmitted signal, g_j is the channel gain and n_j is the additive noise component. As each received signal r_j contains the transmitted information, it can be argued that the linearly combined composite signal

$$r = r_1 + r_2 + \cdots + r_N = \sum_{j=1}^{N} g_j s + \sum_{j=1}^{N} n_j \tag{5.2}$$

will be a better representation of the received signal, as the transmitted signals will add constructively, while the noise components, being random, will combine destructively [44], leading to a higher local SNR. This method, unfortunately, requires the system to be coherent so that the received signals are co-phased when diversity combining takes place. In general, this can be overcome with the help of channel estimation, which provides the receiver with knowledge of g_j. The channel knowledge can then be used to correct the phase of each received signal sample r_j so that all the components within the above linear combination are co-phased. By taking into account the phase correction, we have

$$r = a_1 r_1 + a_2 r_2 + \cdots + a_N r_N = \sum_{j=1}^{N} a_j r_j, \tag{5.3}$$

where a_j, $j = 1, 2, \ldots, N$ are the combining coefficients computed from the estimation of the channel. The values of these coefficients are dependent on the combining technique.

5.1.1 Selection Combining

SC combines the signal samples from different antennas by only using the strongest received signal sample r_k [44]. It is, strictly speaking, not a combining technique but more of a switching or selection technique. The samples that are not selected are discarded and do not contribute to the formation of r, and hence SC is perhaps

the simplest diversity technique. Let k be the index of the channel whose power $p_k \geq p_{j \neq k}$. The combining coefficients then have values of

$$a_j = \begin{cases} 1, \text{ for } j = k \\ 0, \text{ for } j \neq k. \end{cases} \tag{5.4}$$

Regarding the selection of the best antenna (and its sample), there are several ways to do it. Smallest BER controlled diversity is the most effective, but the most computationally expensive way. In this scheme, a test signal is used to get the received signals from all the antennas, and the antenna whose signal has the fewest errors is used. This is done periodically, especially for fast fading channels, and the antenna with the best BER is chosen dynamically. This method performs well when the channel does not change greatly between two test signals. Another selection method is *received signal strength indication (RSSI)* controlled selection diversity. The antenna with the highest RSSI is chosen by monitoring the field strength at the receive antennas [48, 91]. When the receiver uses channel estimation, the best received sample corresponds to the channel that has the greatest channel gain approximation, which corresponds to the highest approximate local power [54]. It should be noted that SC does not require any knowledge of the received phases, as there is no linear combining involved here. Thus, it can be used with either non-coherent or differentially coherent modulation schemes without the need for phase correction (leading to complex channel estimation scheme) [92].

By assuming a Rayleigh fading channel and AWGN with a variance of 1, the average output power ratio $\bar{p}$ resulting from the application of SC with N antennas is [44, 93]

$$\bar{p}(N) = \sum_{k=1}^{N} \frac{1}{k}. \tag{5.5}$$

According to this, when the N^{th} channel is added, $\bar{p}$ is only increased by $1/N$, for example: $\bar{p}(2) = 1 + 1/2 = 3/2$, $\bar{p}(3) = 1 + 1/2 + 1/3 = 11/6$, etc. Thus, when $N > 4$, there is no significant change in the average local power ratio due to further increasing the number of channels. In other words, the performance of SC does not improve greatly when more than 4 antennas are used.

5.1.2 Equal Gain Combining

In EGC, all channels are assigned equal gain. In other words, EGC simply adds together the received samples r_j, giving each antenna unity weight for a N channel EGC system [44]. Hence, for a coherent receiver,

$$a_j = 1, j = 1, 2, \ldots, N. \tag{5.6}$$

In the case of a non-coherent system, the scheme becomes more complex as the process of phase correction must be carried out before the linear combining. This means that channel phase estimation is required. Let $\hat{\varphi}_j$ be the estimated phase for channel j. The coefficients can be defined as

$$a_j = e^{-\hat{\varphi}_j}, j = 1, 2, \ldots, N. \tag{5.7}$$

So it is clear that EGC is not a good choice for non-coherent system.

By having the assumptions of a Rayleigh fading channel and AWGN with a variance of 1, the EGC technique provides an average power ratio $\bar{p}$ of [44, 93]

$$\bar{p}(N) = 1 + (N - 1)R^2, \tag{5.8}$$

where R is a dimensionless constant which varies with the channel model. For a Rayleigh distribution, $R^2 = \pi/4 \approx 0.785$. The above equation suggests that $\bar{p}(N)$ is linearly proportional to the number of channels. Therefore, increasing the number of antennas will improve the performance of EGC more than SC.

5.1.3 Maximal Ratio Combining

In terms of combining technique, MRC is an optimal scheme since it is based on the likelihood function arguments and is closely related to maximum likelihood detection [94]. It is assumed that the AWGN power is approximately the same for all antennas. In order to achieve the maximum local power output ratio, the linear combining process needs to put more weight on the samples whose SNRs are high and reduce the effects of those with low SNRs. Therefore, the weight assigned to each channel must be proportional to the local root mean squared value of the channel gain, $|g|$, and it is also inversely proportional to the noise variance in that channel, σ^2 [44]. In addition, like the previous EGC scheme, MRC also needs to take into account

the channel phase and delay to ensure the signals are coherently summed. These requirements mean that MRC coefficients need to be computed as ratios of the complex conjugate of their respective fading gains and the noise variance at each antenna [95, 96]. So

$$a_j = \frac{\hat{g}_j^*}{\sigma^2}, \tag{5.9}$$

where $\hat{g}_j^*$ is the complex conjugate of the estimation for channel j gain. Because of the requirement for an accurate channel estimation, MRC is the most complex among the three combining methods.

Using the standard assumption of $\sigma^2 = 1$ as in the previous cases, MRC can deliver an average power ratio of [44, 93]

$$\bar{p}(N) = N. \tag{5.10}$$

It is clear from (5.10) that MRC has the greatest performance improvement with an increase number of antennas.

5.2 Low Complexity Diversity Method

5.2.1 Suitability of Selection Combining Scheme

From the above, it can be seen that MRC is the best scheme for gaining performance improvement with diversity. However, the system here is non-coherent, which means that in order to implement MRC (or EGC), channel estimation must also be included. In addition, the channel that we are considering is fast fading so the channel estimation scheme must adapt to the rapid changes of the complex gain. This means that the channel may need to be estimated for every received signal sample within a transmission frame. For a standard receiver, such a channel estimation scheme can be achieved with the aid of the PSP technique along with a trellis detection process like the implementation in [93]. Unfortunately, the predictive demodulator here proves to be very difficult to incorporate with PSP or any other adaptive channel approximation methods. This is because the detection metric is in the form of a prediction error instead of the difference between the received sample and a hypothesized sample (since the prediction coefficients are pre-computed for a particular channel setting). A solution for this may require a major modification of

the receiver design and significantly increase the detection complexity, which goes against our goal of developing a simple low complexity scheme.

From the above discussion, it is clear that EGC and MRC may not be suitable choices for this particular system. Therefore, SC becomes the only diversity combining technique that is considered here. The advantage of SC is that it does not require a complicated adaptive scheme for accurate channel gain estimations or phase coherence. Its operation only requires information about the current local channel powers, which are much easier to obtain. On the other hand, since this research considers the case of diversity with 2 antennas, according to (5.5), (5.8) and (5.10), the power ratios $\bar{p}(2)$ of SC, EGC and MRC are 1.5, 1.785 and 2. As the values are not very far apart from each other, for a diversity order of 2, there may not be much difference between the performances of the three combining methods. This further supports the suitability of SC for the predictive receiver structure.

5.2.2 Sequence Selection Combining

In this section, a new spatial diversity scheme based on SC for the predictive receiver is proposed. Recall from Chapter 3 that a branch metric is computed from a set of the $KN_{ss} + 1$ latest contiguous received samples. This means that the traditional approach of simply selecting a received sample that has the highest signal energy for each step cannot be applied. This is because the samples received by different antennas are out of phase. So, when the best samples from different antennas (selected at different time steps) are arranged to form the required sample sequence for a prediction, the phase of the resulting sequence is not continuous and its corresponding prediction degrades significantly causing a large increase in BER. Therefore, instead of selecting the best sample, SC used here selects the best sample sequence at the current time step.

A simple method to select the best sequence was developed by considering the average power of the samples within each sequence. In order to do it, one can simply compute the absolute value for each sample within a sequence, then calculate the mean value from these absolute values as a representation of the average power of that sequence. The sequence that has the highest mean value is likely to have better received samples, and thus provide a better prediction. Unfortunately, the

considered channel is fast fading, which means the complex channel gain of the current sample may be quite different from that of the oldest samples within the sequence. Therefore, a sequence, though having the strongest recently received samples, may still be eliminated if its older samples are weak leading to a low average absolute value. Given that the effects of the recent samples are higher than those of the older ones in linear prediction, discarding of such sequences may result in a performance penalty for the diversity scheme. Hence, instead of considering every sample within a sequence, only the k most recent samples are involved in the process. The number can be affected by many factors such as the modulation, prediction order, number of samples in a symbol period and channel model. Thus, it may be best to determine k via simulation. Figure. 5.1 summarizes the operation of the developed *sequence selection combining (SSC)* scheme for M receive antennas diversity. The selection metric for a sample sequence $y^i_{nN_{ss}}, ..., y^i_{(n-K)N_{ss}}$ received from antenna i is calculated from the k most recent samples as

$$Metric = E\{|y^i_{nN_{ss}}|, ..., |y^i_{(n-K)N_{ss}}|\} = \frac{|y^i_{nN_{ss}}| + ... + |y^i_{(n-K)N_{ss}}|}{k}, \tag{5.11}$$

where $|.|$ denote the compute absolute value operation.

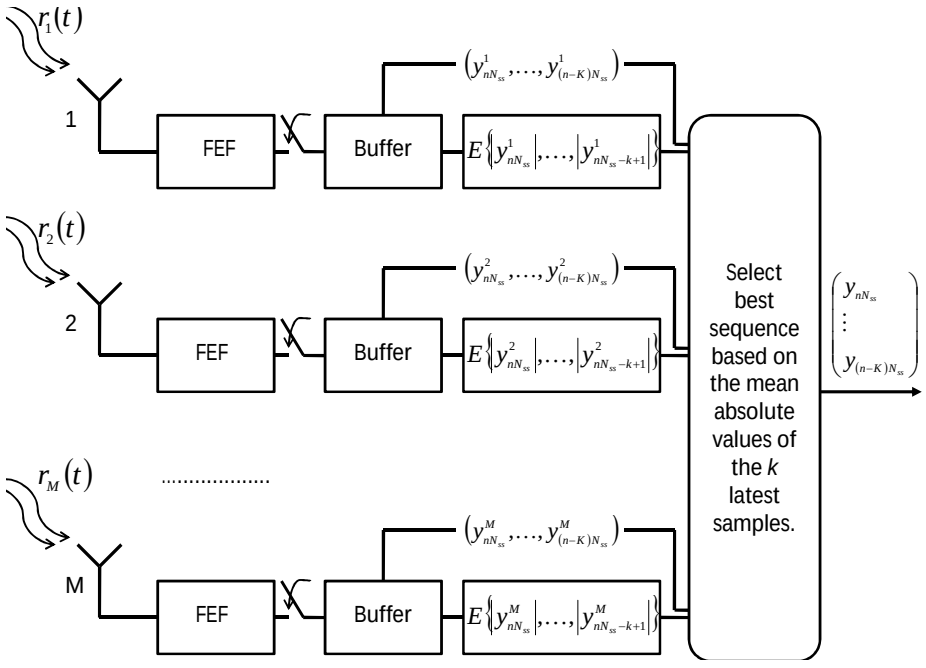
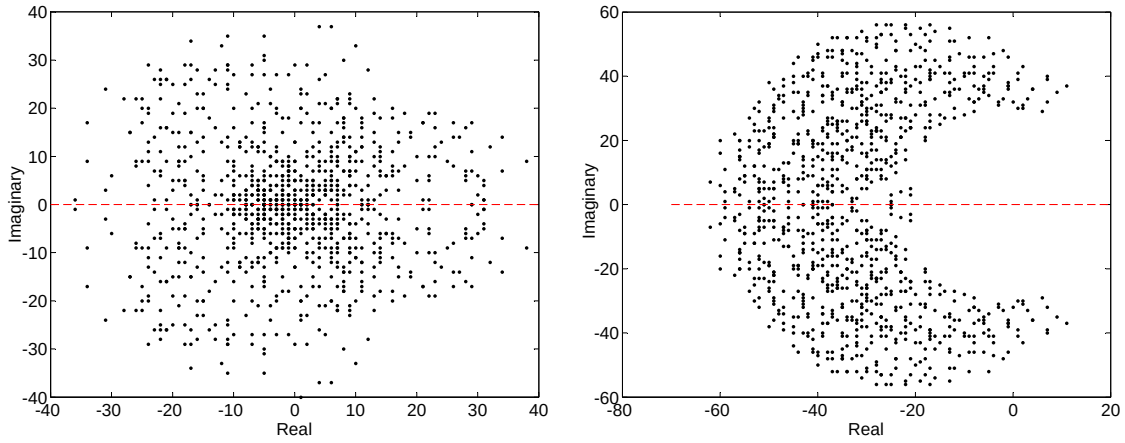


Figure 5.1: SSC diversity.

5.3 Predictive Coefficient Symmetry

As discussed in Chapter 3, the construction of LUT in the receiver design is a standard method for avoiding the need for direct channel adaptivity, thus leading to a significant complexity reduction. It was found out in this research that the current utilization of the LUT in the receiver design can be improved. Such improvement is due to a symmetry on the real axis within these coefficient LUTs. For example, let us consider the coefficient LUT used in Chapter 4 simulations. Because the LUT is constructed for $N_{ss} = 2$, it contains all the possible values for two distinct prediction filters, $\boldsymbol{w}_0$ and $\boldsymbol{w}_1$. As the LUT is configured for $N_{ss} = 2$ and $K = 3$, there are 6 coefficients in $\boldsymbol{w}_0$ and 5 coefficients in $\boldsymbol{w}_1$.



(a) First coefficient of $\boldsymbol{w}_0$.

(b) Third coefficient of $\boldsymbol{w}_1$.

Figure 5.2: Scatter plots of coefficient sets corresponding to a LUT that has settings of $N_{ss} = 2$, $K = 3$, $L = 3$, $f_{d_{max}} = 50\ Hz\ (0.014T)$, $\tau = 12.5\ \mu s$ and $\Upsilon = 30\ dB$.

Figure 5.2 (a) shows the scatter plot of all the possibilities for the first coefficient of $\boldsymbol{w}_0$ contained in the LUT. It can be seen that the real axis divides the coefficient set into two groups that are the reflections of each other. This is also observed in the third coefficient of $\boldsymbol{w}_1$ shown in Fig. 5.2 (b). Such symmetry not only appears in this particular LUT, we found it is also a characteristic of many other LUTs with different settings. In such cases, the memory requirement for storing the predictive coefficient LUT can be effectively halved. Given that the storage of a coefficient LUT is a major memory usage in the construction of the demodulator, this symme-

try allows a considerable reduction in the total memory requirement, especially in the case of a multiple samples per symbol system or when using multiple LUTs in the demodulation process.

Here, I will show a general proof that the symmetry exists for all the LUTs generated for this particular receiver. We consider an A-ary CPM modulation whose constellation is $\Omega_a = \Big\{ \pm 1, \pm 3, ..., \pm A-1 \Big\}$. The group of all possible symbol sequences whose length is fixed can be divided into two sub-groups, $\boldsymbol{A}$ and $\boldsymbol{B}$, in which for every sequence $\boldsymbol{a} \in \boldsymbol{A}$, there is always a corresponding sequence $\boldsymbol{b} \in \boldsymbol{B}$ such that $\boldsymbol{b} = -\boldsymbol{a}$. Let the length of these sequences be $K + L - 1$. According to (3.36) and (3.37), we have

$$\begin{aligned} s(t, \boldsymbol{b}) &= \exp\left(j2\pi h \sum_{k=n-K-L+1}^{n-1} b_k q(t - kT) \right) && (5.12) \\ &= \exp\left(- j2\pi h \sum_{k=n-K-L+1}^{n-1} a_k q(t - kT) \right) && (5.13) \\ &= s^*(t, \boldsymbol{a}). && (5.14) \end{aligned}$$

Note: $\frac{E_s}{T}$ can be left out in the above equations as it is just a constant. Therefore,

$$s\left\{ \left(n - \frac{i}{N_{ss}} \right) T - \alpha - \tau, \boldsymbol{b} \right\} = s^*\left\{ \left(n - \frac{i}{N_{ss}} \right) T - \alpha - \tau, \boldsymbol{a} \right\}. \quad (5.15)$$

By substituting the above expression into (3.41), $V_x(i_1, i_2|\boldsymbol{b})$ can be expressed as

$$\begin{aligned} V_x(i_1, i_2|\boldsymbol{b}) &= \sum_{l=0}^{N_p} \sigma_l^2 \int_{-\infty}^{\infty} \overline{s\left\{ \left(n - \frac{i_1}{N_{ss}} \right) T - \alpha_1 - \tau_{l_1}, \boldsymbol{a} \right\}} h_f(\alpha_1) d\alpha_1 \\ &\times \int_{-\infty}^{\infty} \overline{s^*\left\{ \left(n - \frac{i_2}{N_{ss}} \right) T - \alpha_2 - \tau_{l_2}, \boldsymbol{a} \right\}} h_f^*(\alpha_2) d\alpha_2 \qquad (5.16) \\ &\times J_0\left(2\pi f_{d_{max}} \frac{(i_2 - i_1)T}{N_{ss}} \right), \end{aligned}$$

where N_p is the number of paths within the channel and the operator $\bar{\cdot}$ indicates the complex conjugate. Since the impulse response $h_f(t)$ is real, by applying the complex equality $\overline{z} + \overline{w} = \overline{z + w}$ and considering integration as a form of linear

addition, we have

$$\begin{aligned} V_x(i_1, i_2|\boldsymbol{b}) &= \sum_{l=0}^{N_p} \sigma_l^2 \overline{\int_{-\infty}^{\infty} s\left\{\left(n - \frac{i_1}{N_{ss}}\right)T - \alpha_1 - \tau_{l_1}, \boldsymbol{a}\right\} h_f(\alpha_1) d\alpha_1} \\ &\times \overline{\int_{-\infty}^{\infty} s^*\left\{\left(n - \frac{i_2}{N_{ss}}\right)T - \alpha_2 - \tau_{l_2}, \boldsymbol{a}\right\} h_f^*(\alpha_2) d\alpha_2} \\ &\times J_0\left(2\pi f_{d_{max}} \frac{(i_2 - i_1)T}{N_{ss}}\right). \end{aligned} \tag{5.17}$$

As the Bessel function output is a real number, we can first apply the identity $\overline{z} \times \overline{w} = \overline{z \times w}$ to the right hand side of (5.17) and then utilize $\overline{z} + \overline{w} = \overline{z + w}$ for the complex summation of different channel paths. The obtained result is

$$\begin{aligned} V_x(i_1, i_2|\boldsymbol{b}) &= \left[\sum_{l=0}^{N_p} \sigma_l^2 \int_{-\infty}^{\infty} s\left\{\left(n - \frac{i_1}{N_{ss}}\right)T - \alpha_1 - \tau_{l_1}, \boldsymbol{a}\right\} h_f(\alpha_1) d\alpha_1 \right. \\ &\times \int_{-\infty}^{\infty} s^*\left\{\left(n - \frac{i_2}{N_{ss}}\right)T - \alpha_2 - \tau_{l_2}, \boldsymbol{a}\right\} h_f^*(\alpha_2) d\alpha_2 \\ &\times \left. J_0\left(2\pi f_{d_{max}} \frac{(i_2 - i_1)T}{N_{ss}}\right)\right]^* \\ &= V_x^*(i_1, i_2|\boldsymbol{a}). \end{aligned} \tag{5.18}$$

From (3.33) and (5.18), it is clear that

$$V_y(i_1, i_2|\boldsymbol{b}) = V_y^*(i_1, i_2|\boldsymbol{a}) \tag{5.19}$$

since $R_f(t)$ is real.

The equality in (5.19), when applied to (3.32) and (3.31), yields

$$\boldsymbol{R}_i(\boldsymbol{b}) = \overline{\boldsymbol{R}_i(\boldsymbol{a})} \tag{5.20}$$

and

$$\boldsymbol{p}_i(\boldsymbol{b}) = \overline{\boldsymbol{p}_i(\boldsymbol{a})}. \tag{5.21}$$

On the other hand, from (3.24) we have

$$\boldsymbol{R}_i(\boldsymbol{a})\boldsymbol{w}_i(\boldsymbol{a}) = \boldsymbol{p}_i(\boldsymbol{a}), \tag{5.22}$$

$$\boldsymbol{R}_i(\boldsymbol{b})\boldsymbol{w}_i(\boldsymbol{b}) = \boldsymbol{p}_i(\boldsymbol{b}). \tag{5.23}$$

By using the previously obtained results, equation (5.23) can be re-written as

$$\overline{\boldsymbol{R}_i(\boldsymbol{a})}\boldsymbol{w}_i(\boldsymbol{b}) = \overline{\boldsymbol{p}_i(\boldsymbol{a})}. \tag{5.24}$$

Therefore,

$$\overline{\overline{\boldsymbol{R}_i(\boldsymbol{a})}\boldsymbol{w}_i(\boldsymbol{b})} = \boldsymbol{R}_i(\boldsymbol{a})\boldsymbol{w}_i(\boldsymbol{a}), \tag{5.25}$$

which can be simplified to

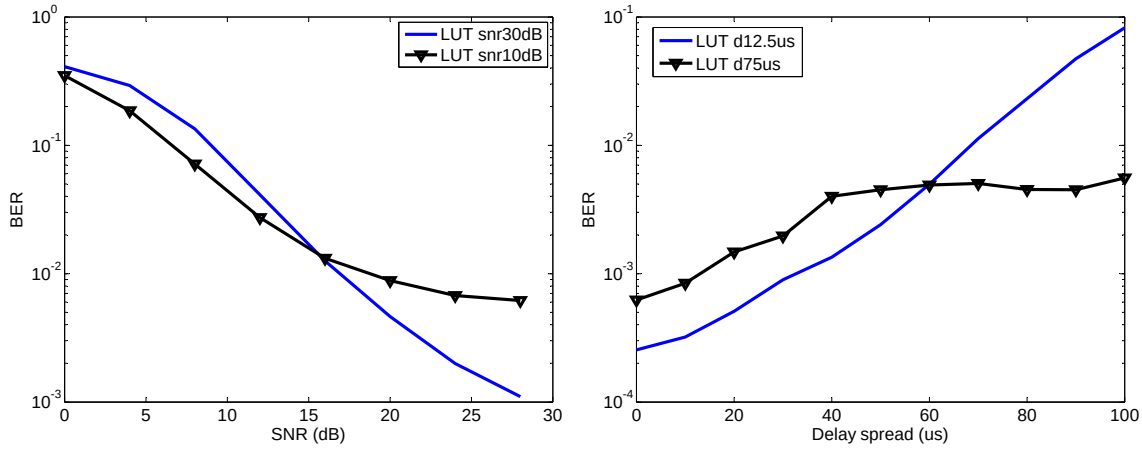
$$\overline{\boldsymbol{w}_i(\boldsymbol{b})} = \boldsymbol{w}_i(\boldsymbol{a}). \tag{5.26}$$

The result in (5.26) means that half the coefficient sets are just the complex conjugate of the other half regardless of the channel settings used for the coefficient computation. The proof is now completed.

5.4 Predictive Demodulation with Multiple Coefficient LUTs

As shown in the simulation results of Chapter 4, the channel delay spread and AWGN level have great influence on receiver performance. Detection accuracy degrades severely as the delay spread becomes high or the noise energy dominates that of the transmit signal. As mentioned earlier, a possible method to improve the demodulation process is to apply a coefficient LUT configured specifically for high delay spread and/or low SNR condition. However, the improvement in detection reliability comes at the cost of performance degradation at low delay spread and/or high SNR condition.

Figure 5.3 shows the hard decision BER curves for different LUTs whose settings differ in delay spread and SNR (the rate 1/2 code is used here). We can see that there is a significant loss of performance at high SNR if the LUT parameter is for low SNR conditions. Likewise, using a LUT with the wrong delay spread is also costly in terms of performance, especially at high delay spreads. Therefore, applying a high delay spread or low SNR calculated LUT is not a good method for improving the average receiver operation.



(a) LUTs with different SNR settings.

(b) LUTs with different delay spread settings.

Figure 5.3: BER curves for a rate 1/2 code with different LUTs. All the above LUTs are built with common configuration parameters of $N_{ss} = 2$, $K = 3$, $L = 3$, $f_{d_{max}}T = 0.014$. (a) Both LUTs are configured with $\tau = 12.5\ \mu s$, but $\Upsilon = 30\ dB$ for the first LUT and $\Upsilon = 10\ dB$ for the second LUT. Transmission channel has $f_{d_{max}} = 50\ Hz$ and $\tau = 30\ \mu s$. (b) Both LUTs are configured with $\Upsilon = 30\ dB$, but for the first LUT $\tau = 12.5\ \mu s$ and $\tau = 75\ \mu s$ for the second LUT. Transmission channel has $f_{d_{max}} = 50\ Hz$ and $\Upsilon = 30\ dB$.

A possible solution for the high delay spread case is to apply a channel monitoring process that can approximate the channel delay spread before the demodulation starts such as the algorithms in [97, 98, 99]. Then depending on the estimated delay spread value, the most suitable LUT (among those stored) can be selected for detection. Such a method, however, does not utilize the pre-existing predictive design very well as it relies on an external structure for selecting the best LUT. Since one of the important aspects of this research is to improve the capability of the predictive design, this method is not investigated here. A more suitable approach is to modify the demodulator design so that more than one LUT is involved in its operation and the achieved overall performance can be the sum of the best performance part for each involved LUT. In other words, having alternative coefficient vectors available for demodulation is a way of substituting memory for adaptivity. For example, in terms of SNR performances in Fig. 5.3 (a), the method aims to combine the lower part of the black BER curve with the higher part of the blue BER curve. In the case of delay spread performance in Fig. 5.3 (b), the combination of the lower part

of the blue BER curve and the higher part of the black BER curve is desirable.

The present research focuses on the use of two different coefficient LUTs for enhancing system performance in SNR and delay spread. This is because the usage of three or more LUTs may not only cause the demodulation process to be very complex, but the performance gain may also be insufficiently different to that of the two LUT case.

5.4.1 LUTs with Different SNR Settings

This section describes a method for using two distinct LUTs whose setting parameters are the same apart from the SNR. The scheme is based on the idea of computing parallel path metrics (each corresponding to a coefficient LUT) for each processing state in the demodulation trellis and combining them to select the overall survivor path. Its operation can be summarized as: For each time step,

1. Compute the parallel branch metrics corresponding to all the LUTs.

2. Add the branch metric corresponding to LUT i to the survivor path metric calculated by LUT i from the previous time step. A new path metric is computed for each LUT.

3. For each transition within the time step, compute ratios ε_1 and ε_2 between the newly computed path metrics and the sum of them, where

$$\varepsilon_1 = \frac{metric_1}{metric_1 + metric_2}, \tag{5.27}$$

$$\varepsilon_2 = \frac{metric_2}{metric_1 + metric_2}. \tag{5.28}$$

4. Multiply the ε_1 by the new LUT 2 path metric, and the ε_2 by the new LUT 1 path metric before summing a weighted version of them together to form the combined path metric of the transition.

5. Compare the combined path metrics corresponding to all the transitions to the current considered state, select the best one and store the (un-scaled) path metric of each LUT that contributes to the chosen combined path metric.

Note: In step 4 above, the path metrics need to be scaled with ε_1 and ε_2 before summation to mitigate the effect of the size difference between predictions of different LUTs. It was found that for the same sequence of predictive samples, the higher SNR setting LUT produces a considerably higher branch metric and thus, higher path metric given that all other settings of the LUTs are the same and they are both normalized. This means that when combining without applying scaling, the path metric corresponding to the higher SNR setting LUT can dominate the resulting overall path metric leading to smaller contribution from the lower SNR setting LUT.

An example for the above process is illustrated in Fig. 5.4 and shows the contention for the survivor path at a particular state in the demodulation trellis. Let $M_n^m(pst)$ be the survivor path metric of the previous state pst that is computed by LUT m ($m = 1, 2$) at time nT, μ_{pst}^m is the branch metric corresponds to LUT m and the transition $[pst, st]$, and $M_n^m(st)$ is the survivor path metric of the current state st.

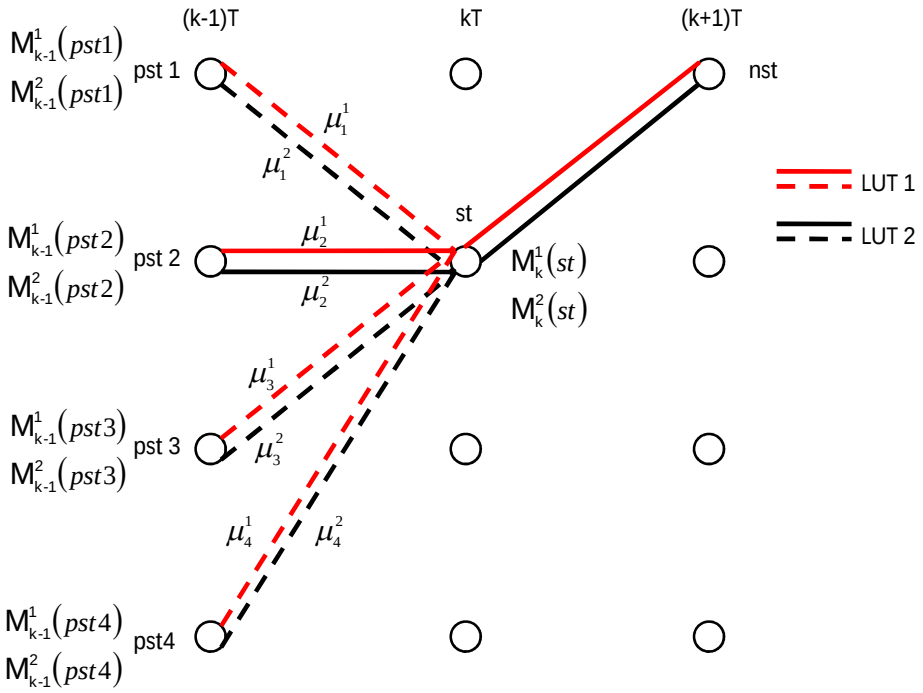


Figure 5.4: Example of demodulation with two different SNR setting LUTs.

Let us consider the transition $[pst, st]$, the path metrics computed for this transition by LUT 1 and LUT 2 are $M^1(pst1 \rightarrow st) = M^1_{k-1}(pst1) + \mu^1_1$ and $M^2(pst1 \rightarrow st) = M^2_{k-1}(pst1) + \mu^2_1$. To calculate the combined path metric for the transition, the following ratios are required:

$$\varepsilon_1 = \frac{M^1(pst1 \rightarrow st)}{M^1(pst1 \rightarrow st) + M^2(pst1 \rightarrow st)}, \tag{5.29}$$

$$\varepsilon_2 = \frac{M^2(pst1 \rightarrow st)}{M^1(pst1 \rightarrow st) + M^2(pst1 \rightarrow st)}. \tag{5.30}$$

By assuming that the SNR parameter of LUT 1 is higher than that of LUT 2, the combined path metric can be computed as

$$M(pst1 \rightarrow st) = \varepsilon_2 M^1(pst1 \rightarrow st) + \varepsilon_1 M^2(pst1 \rightarrow st) \tag{5.31}$$

$$= \frac{2M^1(pst1 \rightarrow st)M^2(pst1 \rightarrow st)}{M^1(pst1 \rightarrow st) + M^2(pst1 \rightarrow st)}. \tag{5.32}$$

Once all 4 combined path metrics are computed, the process selects the smallest one as the survivor metric, say $M(pst2 \rightarrow st)$ in this case. Then the path metrics $M^1(pst2 \rightarrow st)$ and $M^2(pst2 \rightarrow st)$ are stored as the survivor path metrics at state st. In other words, $M^1_k = M^1(pst2 \rightarrow st)$ and $M^2_k = M^2(pst2 \rightarrow st)$. The same procedure can be done for the other transitions within the trellis until the detection process finishes.

Since the LUT combining algorithm relies on the path metric to form the overall metric for selecting the survivor path, it may not be possible to incorporate the BCJR-based algorithm here for soft information calculation. This is because the BCJR-based algorithm calculation is based on branch metrics at each time step and these metrics are not combined here. Therefore, only reliability generation algorithms whose operations are based on path metrics such as the SOVA-based algorithm can be used with this modified demodulator.

5.4.2 LUTs with Different Delay Spread Settings

The procedure for combining the performances of two LUTs whose settings only differ in delay spread parameter is very similar to the one in the last section. The only difference is in the computation for the combined path metric as the higher

delay spread setting LUT tends to (but not always) produce higher branch metrics than the other one and the difference between the branch metrics are often not so high. Therefore, the process for combining LUTs in the last section is modified to be (assuming that LUT 2 has higher delay spread setting):

1. Compute the parallel branch metrics corresponding to all the LUTs.

2. Add the the branch metric corresponding to LUT i to the survivor path metric calculated by LUT i from the previous time step. A new path metric is computed for each LUT.

3. Compute all the parallel path metrics corresponding to all the transitions to the current state. For each parallel pair, compute ratios ε_1 and ε_2 between the path metrics and the sum of them.

4. Select the smallest path metric. If the chosen one corresponds to LUT 1,

   - For each pair of path metrics, multiply ε_1 by the metric that corresponds to LUT 1 and ε_2 by the one that corresponds to LUT 2 before summing the results to form the combined path metric for the corresponding transition,

   $$metric = \varepsilon_1 metric_1 + \varepsilon_2 metric_2. \tag{5.33}$$

   On the other hand, if the smallest path metric is generated from LUT 2,

   - For each pair of path metric, multiply ε_2 by the metric that corresponds to LUT 1 and ε_1 by the one that corresponds to LUT 2 before summing the results to form the combined path metric for the corresponding transition,

   $$metric = \varepsilon_2 metric_1 + \varepsilon_1 metric_2. \tag{5.34}$$

5. Compare the combined path metrics corresponding to all the transitions at the current considered state, select the best one and store the (un-scaled) path metric of each LUT that contributes to the chosen combined path metric.

Let us re-use the example in Fig. 5.4 while still keeping our assumption that LUT 1 has smaller delay spread setting than that of LUT 2. Let a parallel path metric corresponding to a transition j be $M^i(pstj \rightarrow st)$ and its scaling ratio be ε_i^j (where $i = 1, 2$ and $j = 1, 2, 3, 4$). When all the parallel path metrics and ratios

are computed, we consider $min(M^i(pstj \rightarrow st))$. If $min(M^i(pstj \rightarrow st))$ has $i = 1$, then the combined path metric is

$$M(pstj \rightarrow st) = \varepsilon_1^j M^1(pstj \rightarrow st) + \varepsilon_2^j M^2(pstj \rightarrow st) \quad (5.35)$$

$$= \frac{(M^1(pstj \rightarrow st))^2 + (M^2(pstj \rightarrow st))^2}{M^1(pstj \rightarrow st) + M^2(pstj \rightarrow st)}. \quad (5.36)$$

On the other hand, if $i = 2$, then

$$M(pstj \rightarrow st) = \varepsilon_2^j M^1(pstj \rightarrow st) + \varepsilon_1^j M^2(pstj \rightarrow st) \quad (5.37)$$

$$= \frac{2M^1(pstj \rightarrow st)M^2(pstj \rightarrow st)}{M^1(pstj \rightarrow st) + M^2(pstj \rightarrow st)}. \quad (5.38)$$

The rest of the process is quite similar to the other example, which consists of the survivor selection process and storing the parallel path metric corresponding to the survivor path.

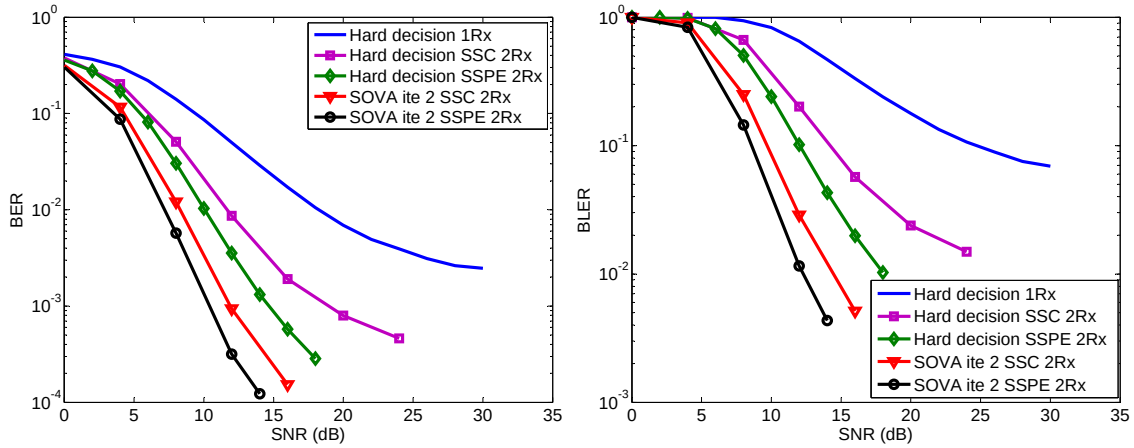
Similar to the algorithm described in Section 5.4.1, the combining algorithm cannot be used with any soft output algorithms that rely on branch metrics. Only path metric based algorithms are able to compute soft output in the modified receiver.

5.5 Simulation Results

The performance of the new diversity scheme and multiple coefficient LUT demodulation technique are simulated over a two path equal power Rayleigh channel model as in Chapter 4. The maximum Doppler shift is fixed at $f_{d_{max}} = 50\ Hz\ (0.014T)$ for reflecting a fast fading environment, which is often the case in public safety operations, while the SNR or channel delay spread is varied depending on the simulation. All the simulations are carried out with the rate 1/2 code and the demodulator is configured to have a 16 state demodulation trellis or $P = 2$. It should be noted that the coefficient symmetry property is applied to all the simulations presented in this chapter. This means that the demodulator only has direct access to half the number of coefficient sets. It was confirmed by simulation that there is no change in the demodulator performance when compared to the result obtained in Chapter 4.

5.5.1 SSC Diversity

In order to determine the SNR performance of the new diversity scheme, simulations were carried out with a $50us$ delay spread channel as this corresponds to a reasonably large cell multicast network. The coefficient LUT used in Chapter 4 simulations is also applied here. With such a configuration for the demodulator, it was discovered that the SSC scheme operates best when the sequence selection process only considers the 3 latest received samples at every time step. Hence, all the simulation results for the new SSC diversity scheme are obtained for $k = 3$.



(a) BER Result.

(b) BLER Result.

Figure 5.5: SNR performance of SSC diversity, $\tau = 50\ \mu s$ and $f_{d_{max}} = 50\ Hz$.

Figure. 5.5 compares the performance of the new diversity scheme with that of the standard hard decision receiver without diversity and the original SSPE diversity described in Chapter 3 [13]. We can see that the performances of both diversity methods are clearly superior to that of the detector in which diversity is absent. This matches theoretical expectations. In a comparison between the two diversity techniques, it can be seen that the SSPE produces better performances for both hard and iterative detection. Table 5.1 displays the SNR improvement differences between the two techniques with respect to SSC diversity performance with hard detection. These values indicate that the new diversity technique might not be able to replace the original one for hard decision detection. This is because at low BER, the SSPE scheme provides a very high SNR gain over that of the new scheme, for example, 4.29 dB at 0.1% BER. Although the SSC method approximately halves

the demodulator complexity for introducing diversity, the losses in SNR performance may be too high for such a trade off. In addition, hard decision detection itself is not a complex process and thus, the system is able to handle the more complex SSPE technique without causing much delay.

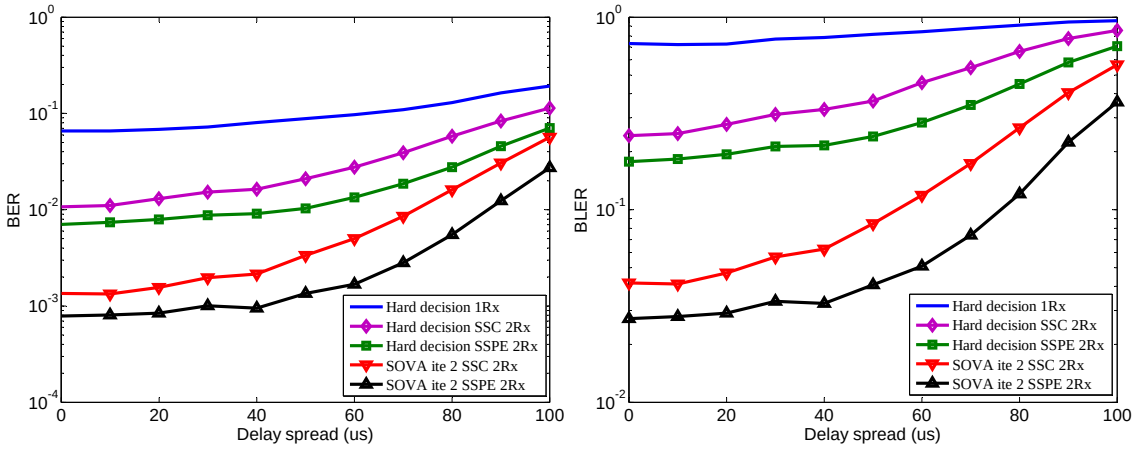
Table 5.1: SNR performance comparison for different diversity schemes at different BERs.

	SSPE-SOVA ite 2	SSC-SOVA ite 2	SSPE-Hard	SSC-Hard
10% BER	2.47 dB	1.76 dB	0.59 dB	0 dB
1% BER	4.49 dB	3.38 dB	1.61 dB	0 dB
0.1% BER	8.54 dB	7.06 dB	4.29 dB	0 dB

The situation, however, changes when iterative detection is applied to the system. Like the case of single antenna reception in Chapter 4, iterative detection has proven to be a very effective method for enhancing the performances of these diversity schemes, especially for the SSC. It is shown in Table 5.1 that SOVA iterative detection introduces a SNR gain of 4.25 dB for a system utilizing SSPE diversity and a gain of 7.06 dB in the case of SSC diversity over hard detection. As there is such a massive gain for SSC, the difference between the two diversity techniques are now only 1.48 dB at 0.1% BER and this value only increases slowly as BER decreases. Given that the implementation of iterative detection is relatively complex even with the simple SOVA-based algorithm, complexity starts to be an issue when diversity is employed. Since demodulation is the most computationally demanding operation in the system and it is repeated twice here, the SSC scheme can allow a very significant complexity reduction in the reception process. Therefore, SSC has an advantage over SSPE here.

Delay spread performances of these diversity schemes are captured in Fig. 5.6. Simulation results were obtained with a very low SNR setting of 10 dB, to test whether the diversity and iterative techniques can allow the receiver to operate in a multipath fast fading channel with such a low signal strength. The figure clearly shows that the standard hard detection with single antenna is unable to operate in such conditions even for flat fading. On the other hand, the application of diversity allows a BER of 1% or lower (required for reliable voice transmission) to be achieved

with a maximum delay spread of approximately 40 μs for SSC and 60 μs for SSPE. The addition of iterative techniques not only increases the delay spread range to 80 μs for SSC and 90 μs for SSPE but also reduces their error floors at low delay spread from approximately 1% to 0.1%. This again demonstrates the effectiveness of iterative detection and shows that SSC offers an attractive performance versus complexity trade off (as its delay spread performance is only 10 μs behind that of the original scheme). It is worth noting that for both types of detection, there is not much difference in the performances of the diversity techniques between flat fading and delay spread of 40 μs.



(a) BER Result.

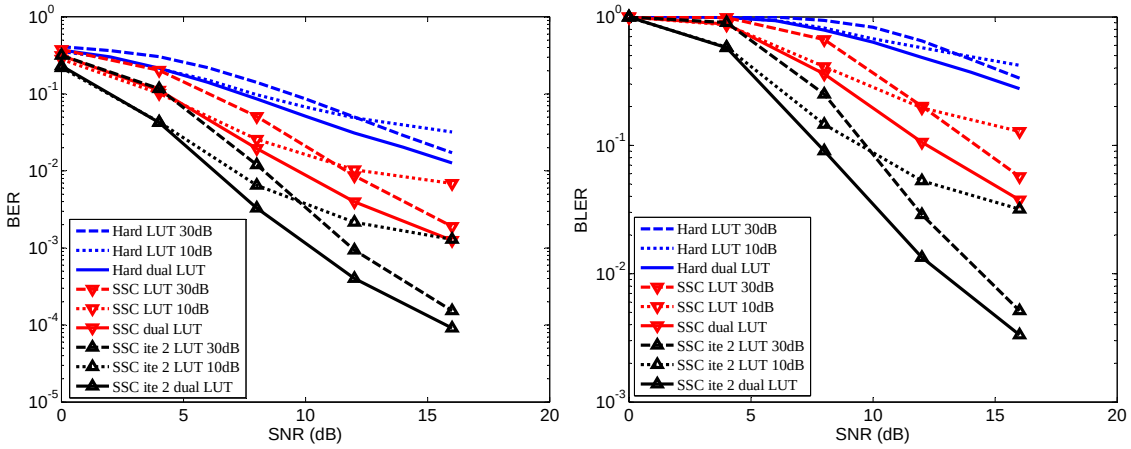
(b) BLER Result.

Figure 5.6: Delay spread performance of SSC diversity, $\Upsilon = 10\ dB$ and $f_{d_{max}} = 50\ Hz$.

5.5.2 LUTs with Different SNR Settings

The simulation results for the dual SNR LUT demodulation method are displayed in Fig. 5.7. The two coefficient LUTs used in these simulations are the ones shown in Fig. 5.3 (a). It can be seen from the figure that the developed combining method not only achieves the goal of combining the best performance parts of each LUT but also smooths out the middle SNR region between these parts. As a result of this, the dual LUT receiver has a lower detection error rate in the middle SNR range than the original design that utilizes any one of the chosen LUTs. We can see that there is a considerable SNR improvement when comparing the performance of this scheme

to the traditional approach of demodulation with the most suitable coefficient LUT (which is the 30 dB setting LUT in this case). Here, the channel delay spread is set to $\tau = 50$ μs for the same reason as Section 4.5.1.



(a) BER Result.

(b) BLER Result.

Figure 5.7: SNR performance of dual SNR LUT receiver, channel has $\tau = 50$ μs and $f_{d_{max}} = 50$ Hz.

Table 5.2 displays the SNR gains over the 30 dB setting LUT performances from applying the dual-LUT technique. It shows that the SNR gain is approximately 2 dB across all the detection schemes and slowly reduces as the SNR increases. Such a reduction happens because the two error curves will eventually merge with each other when the SNR becomes high enough. For hard detection, its gain may not be very useful as it happens at high BER. However, this will change with a better transmission channel. It should also be noted that the SNR gains shown here are for the particular pair of coefficient LUTs and will vary over different LUT selection. As the result of computing two parallel branch metrics, the dual LUT demodulation process is approximately twice as complex as the standard one. Therefore, there is a trade off between SNR gain and complexity here.

Note: the SSC diversity technique is the only one considered in these simulations because of its complexity being no more than the single antenna reception case. Therefore, the overall detection process can be less computationally demanding when all the enhancement techniques of SSC diversity, dual LUT and iterative detection

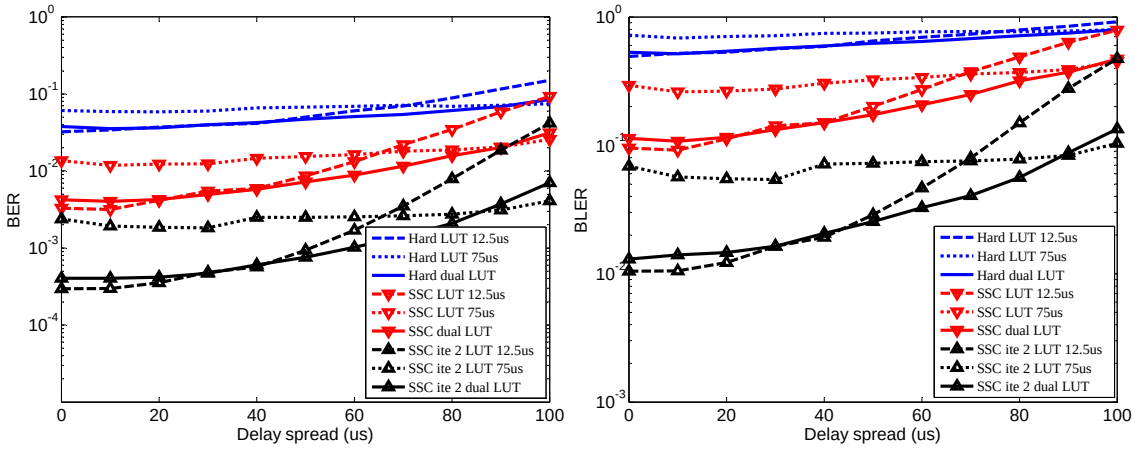
Table 5.2: SNR gains at different BERs. These results are extracted from Fig. 5.7 as the SNR differences between the dual LUT error curves and the LUT 30 dB curve.

	1 Rx Hard 2	SSC Hard 2	SSC SOVA ite 2
10% BER	1.99 dB	1.79 dB	2.28 dB
1% BER	-	1.98 dB	2.03 dB
0.1% BER	-	-	1.62 dB

are applied at the same time.

5.5.3 LUTs with Different Delay Spread Settings

The performance of the dual delay spread LUT technique is obtained for the two coefficient LUTs in Fig. 5.3 (b) whose delay spread settings are 12.5 μs and 75 μs. The results are shown in Fig. 5.8.



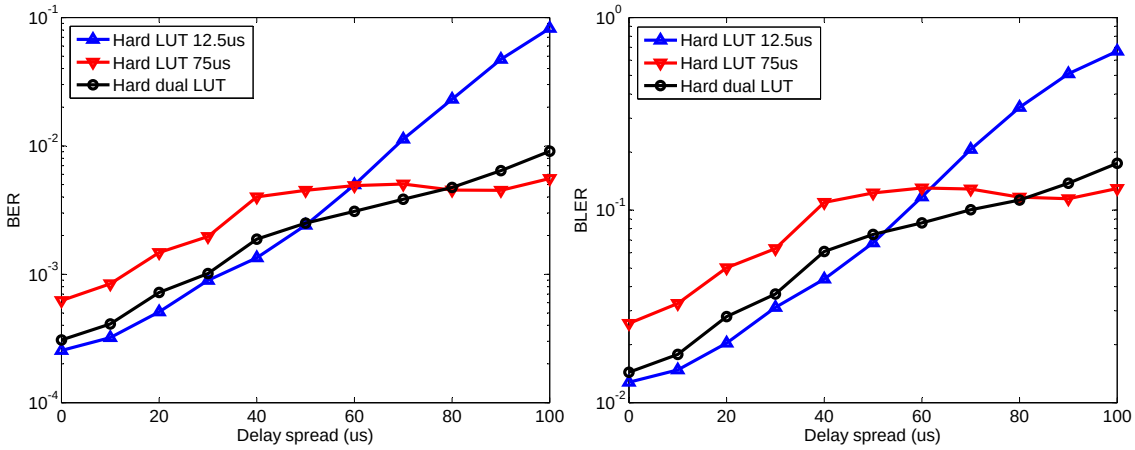
(a) BER Result.

(b) BLER Result.

Figure 5.8: Delay spread performance of dual delay spread LUT receiver, channel has $\Upsilon = 12\ dB$ and $f_{d_{max}} = 50\ Hz$.

We can see that in comparison to the 75 μs, the application of the technique does bring a significant error rate reduction to the system around mid to low delay spread while the LUT performance is better at very high delay spread values. At low end of the delay spread range, a large error rate decrement of approximately a

decade is obtained. Similarly, the 75 μs LUT has a slightly better performance at low delay spread, while being outperformed at mid to high delay spread, especially for iterative detection with diversity. Here, we can observe a delay spread gain of near 20 μs provided by the dual technique. Given that the SNR used in these simulations are relatively low, 12 dB, the error rate can be significantly affected by the AWGN and less dependent on the delay spread condition. Thus, when a higher SNR value is applied, the channel delay spread becomes the main factor causing error in detection so the technique gains can be even higher. This is illustrated in Fig. 5.9 that shows the technique performance with hard detection when $\Upsilon = 30$ dB.



(a) BER Result.

(b) BLER Result.

Figure 5.9: Delay spread performance of dual delay spread LUT receiver, channel has $\Upsilon = 30$ dB and $f_{d_{max}} = 50$ Hz.

5.6 Summary

In this chapter, we have developed a dual diversity technique based on SC that achieves reasonable diversity gain, especially when using iterative detection while its implementation complexity is just slightly higher than that of the single antenna receiver. Given that iterative detection is likely to be a key feature in future mobile devices, this is a good practical design for the predictive demodulator. It was shown in simulations that this scheme performs very well in a fast fading multipath

channel as it is able to improve the error performance by more than 10 dB over a single channel receiver as well as achieving a much lower error rate in delay spread performance. In comparison with the original diversity scheme, it is inferior to the original diversity scheme for hard detection as its low complexity advantage can not compensate for a significant loss in detection accuracy. However, when iterative detection is considered, the simplicity of SSC becomes a critical factor in practical implementation and the SSC detection quality is only 1.5 dB worse than that of the SSPE scheme. Therefore, in such cases, SSC can be a promising choice for replacing the original diversity method in practical implementation. It should also be pointed out that the complexity of the SSC is approximately the same regardless of the number of reception antennas, while the demodulation complexity of SSPE increases linearly with it.

The other contribution here is an enhancement of the demodulation process, which provides better utilization of the predictive coefficient LUTs. This involves a complete proof on the symmetry property of the predictive coefficient sets, which allows the demodulator to decrease its memory storage by half. Also, the demodulation process is also modified so that a pair of coefficient LUTs instead of one LUT can be used in the operation. The combination of two LUTs, whose settings only differ in SNR or delay spread, improves the overall SNR or delay spread performance of the demodulation process by combining the best performance segment of each LUT together. Simulations have shown that for a particular pair of coefficient LUTs, a SNR gain of 2 dB is achieved with such a combination. On the other hand, if the LUTs differ in the assumed delay spread parameter, a significant gain in terms of delay spread tolerance is shown. The disadvantage of this method is the increase in demodulation complexity. The application of dual LUTs effectively doubles the computational load of the demodulator. However, given that the delay spread effect is very hard to mitigate and the technique has shown a significant delay spread improvement, it may be worth putting investment in this technique for later generations of mobile devices that utilize the predictive demodulation.

Chapter 6

Conclusion

In this thesis, we have developed a number of techniques for improving the performance of a promising predictive demodulator for CPM signal in fast fading multipath channel. These improvements consist of the generation of soft information from the demodulation process, thus allowing the system to employ soft or iterative detection. In addition, a very low complexity diversity method that provides reasonably good performance as well as methods for the demodulator to have better utilization of its coefficient LUTs were found. In this chapter, the results obtained are summarized and possible future research directions are described, based on the findings of the study.

6.1 Soft and Iterative Detection

In this thesis, we have developed two soft output algorithms for producing reliability information from the demodulator trellis operation. One algorithm is based on the SOVA framework while the other is a variation of the BCJR algorithm. These algorithms allow the system to exchange both soft and hard information between the demodulator and error control decoder, which is the basis for soft and iterative detection.

In terms of soft detection, both algorithms outperform hard detection. For a fast fading multipath channel whose delay spread is approximately a quarter of the symbol rate, the SOVA-based algorithm gives SNR gains of 2.61 dB and 4.74 dB with the rate 3/4 and rate 1/2 codes while the BCJR-based algorithm achieves 2.31

dB and 6.98 dB respectively at 0.5% BER. The BCJR-based algorithm provides a significant gain at low delay spread that quickly diminishes with the increment of delay spread, while the SOVA-based algorithm has a higher error rate at low delay spread, but its performance does not degrade as fast as the BCJR-based algorithm. In the case of fading performances, for a particular channel condition, the application of these algorithms has lowered the error rate by approximately half a decade for a wide range of fading rates ranging from flat fading to a Doppler spread of 5% of the symbol rate.

A closed-loop structure was developed for iterative detection. As the information available at the output of the error control decoder is for uncoded data while the demodulator operates on coded symbols, the feed back loop is characterized by a soft encoding process. This process takes the soft and hard information generated from the decoder to produce the reliability of each symbol within the modulation constellation which can be used by the demodulator with minor modification in its design.

Simulation results have shown us that most of the performance gain from iterative detection is achieved with two iterations with additional iterations providing little gain. As expected, for the same channel condition, the iterative scheme (with two iterations) nearly doubles the performance gains of SOVA-based soft detection in SNR, delay spread and fading rate. For example, a SNR gain of 7.59 dB is achieved at 0.5% BER instead of 4.74 dB and a delay spread gain of 30 μs while SOVA-based detection only achieves 15 μs. As the soft encoding process complexity depends on the trellis size of the error control code, this design favors simple codes whose trellis sizes are not large.

6.2 Spatial Diversity Technique

Apart from iterative detection, a very effective way of increasing the reliability of the demodulation process is reception diversity. For this particular demodulation design, spatial diversity is of interest. In the present research, a very low complexity diversity scheme that achieves good diversity gain was found. The method is based on standard SC, but significantly differentiates itself from SC as it involves a selection process of a sample sequence instead of an individual sample. The best among

such sequences is chosen by simply considering the average energy of a few of the latest received samples.

Although simple, the technique allows a significant diversity gain. At the BER of 0.3%, a gain of more than 10 dB is achieved over no diversity reception for hard detection. Yet, for low delay spread channels, the diversity technique effectively shifts the error rate down by half a decade. Given that the SNR considered here is only 10 dB (quite low for the multipath channel), thus causing an error floor in the detection process, it is expected that an extensively better delay spread gain can be achieved with higher SNR value.

When compared to the previously developed diversity technique, the new scheme is inferior in terms of reception accuracy. At 0.1% BER, the new technique performance is 4.25 dB less than that of the other one for hard decision, while their difference is only 1.48 dB for SOVA-based iterative. These results indicate that the new technique is inferior to the other in hard detection, but is close in iterative detection. Given that the new SSC technique allows the demodulation process to be only half as complex as that of the previous SSPE technique, it can be a very useful diversity design for replacing the SSPE scheme for iterative reception in which computational load can be an issue in mobile device implementation.

6.3 Coefficient LUT Utilization

It was realized in this research and in [13, 14] that the performance of the predictive demodulator varies significantly when the coefficient LUT is changed. In [14], an attempt was made to identify the best LUT for different channel settings. Although a few LUTs were identified to have very good overall performances, each of these LUTs alone can not allow the receiver to achieve the best possible performance for a given state of the transmission channel, say delay spread or SNR. This happens because a coefficient LUT is restricted by certain channel parameters used in its construction. As a consequence, the receiver utilizing such an LUT can only reach its ideal performance when the actual transmission channel is close to the description given by the LUT parameters.

Since a single LUT performance is quite limited, this research focused on a method that allows the demodulation process to be supported by multiple LUTs. Such a method was found for the case of 2 LUTs whose only difference is in SNR or delay spread parameter (as these are the major factors that affect the receiver performance). This is done by computing the path metric corresponding to each LUT separately and combining them to form the overall path metric for survivor contention at each state in the demodulation trellis. As a result of relying on path metric, the soft information can only be computed by those algorithms that utilize path metrics such as those of the SOVA family. This may be considered as a limitation of this technique.

Simulations have shown that when using pairs of LUTs with SNR settings of 10 dB and 30 dB, the developed method performance is equivalent to the 10 dB LUT and 30 dB LUT at low and high SNR condition, respectively. Yet, it is better than both of their performances for the middle SNR range. Hence, an approximate 2 dB gain was achieved across a range of BERs and this gain does not vary much between hard and iterative detection. In terms of delay spread performance, simulations were run with a pair of LUTs whose delay spread settings are 12.5 μs and 75 μs. The obtained results show that detection accuracy is slightly higher than that of the 12.5 μs and 75 μs LUTs at low and high delay spread values respectively and surpasses both of them over a certain delay spread range. A significant gain of approximately 20 μs can be achieved for a number of detection schemes such as hard detection, diversity detection and iterative detection with diversity.

The advantages introduced by the combining method is not without costs. These are mainly in the form of computational load and memory requirement of the demodulator being effectively doubled. Fortunately, the memory issue was solved by a discovery of a symmetry across the real axis of the predictive coefficient. This means that half coefficients are the conjugates of the other half. A general proof that this is the case for all the coefficient LUTs was given in this thesis. So, only half the number of coefficients are needed to be stored in a given LUT, thus effectively shrinking the LUT size and the memory requirement by nearly 50%. This may prove to be very useful for the dual LUT technique when the system utilizes a large number of samples per symbol that significantly increases the coefficient LUT size.

6.4 Suggestions for Future Research

In this section we give a list of possible areas for further research which should be considered to further enhance the performance of the systems I have developed in this thesis.

- This research is limited to the use of convolutional codes provided in the standard APCO Phase 1. As these codes are not very strong for a multipath fading channel, the work should be extended to investigate the system performances with better codes.

- Derive performance bounds for the predictive receiver with and without diversity for both hard and iterative detection.

- Investigate the performance of the BCJR-based algorithm when applied to iterative detection.

- Extend the SOVA-based algorithm so that its reliability computation can rely on the n best paths, where $n \geq 2$. A value of n greater than 2 may be necessary to generate good soft information when the modulation has a large constellation.

- Combine space-time coding and receiver space diversity for extending the receiver to *multiple input multiple output (MIMO)* transmission, which further improves its performance or data throughput.

- Research if it is possible to extend the idea of the SSC diversity scheme to MRC or EGC version of the sequence combining technique. This may require an investigation into a low complexity technique that can estimate the channel gain or at least its phase.

- Modify the dual LUT technique so that algorithms like the BCJR-based algorithm can compute soft information from the dual LUT demodulation process. This may prove very useful as the BCJR-based algorithm performance is outstanding at low delay spread condition while at the same time, the dual delay spread LUT technique can offer an excellent performance for high delay spread channels.

- Extending the dual LUT technique so that the receiver can have both SNR and delay spread gains at the same time. More than 2 LUTs may be required in such an extension.
- Instead of trying to combine the performances of multiple LUTs, a possible approach to improve the demodulator performance is to have a bank of coefficient LUTs, whose set of channel parameters are distinct from each other, and a method for selecting the most suitable LUT for the current transmission channel. The LUT selection process may require a channel estimation scheme.
- Complexity analysis for the developed techniques.

Bibliography

[1] F. R. Yu, H. Tang and V. C. M. Leung, *QoS Provisioning in Public Safety Radio and Commercial Cellular Integrated Networks for First Responders and Critical Infrastructures*, in Proc. IPCCC, 2007, pp. 570-575.

[2] TIA, *APCO Project 25 System and Standards Definition, TIA/EIA Telecommunications Systems Bulletin*, TSB102-A 1995.

[3] K. Balachandran, K. C. Budka, T. P. Chu, T. L. Doumi, J. H. Kang, Bell Labs and L. Technologies, *Mobile Responder Communication Networks for Public Safety*, IEEE Communications Magazine, Jan. 2006.

[4] J. B. Anderson, T. Aulin and C. W. Sundberg, *Digital Phase Modulation*, Plenum Press, New York, 1986.

[5] C. W. Sundberg, *Continous Phase Modulation*, IEEE Com. Mag., vol.24, pp. 25-38, Apr., 1986.

[6] T. Aulin, N. Rydbeck and C. W. Sundberg, *Continous Phase Modulation - Part I and Part II*, IEEE Trans. Commun., vol. 29, pp. 196-225, Mar. 1981.

[7] Hillebrand and Friedhelm, *GSM and UMTS, The Creation of Global Mobile Communications*, John Wiley & Sons, Dec. 2001.

[8] J. Cavers and P. Green *Applications of Multichannel Receivers*, Internal Report, The NZi3 Wireless Research Centre, Christchurch, Jan. 2009.

[9] J. B. Anderson and C. E. W. Sundberg, *Advances in Constant Envelope Coded Modulation*, IEEE Communications Magazine, vol. 29, pp. 36-45, 1991.

[10] A. Griffin and D. P. Taylor, *On Differentially Demodulated CPFSK*, in Proc. IEEE ICC., vol. 1, pp. 354-358, June 1996.

[11] A. Griffin and D. P. Taylor, *Coding CPFSK for Differential Demodulation*, IEEE Trans. Commun., vol. 48, pp. 721724, May 2000.

[12] R. H. Yang and D. P. Taylor, *Trellis Coded Continous Phase Frequency Shift Keying with Ring Convolutional codes*, IEEE Trans. Infor. Theory, vol. 40, pp. 10571067, July 1994.

[13] J. Cavers, *Predictive Receivers for Frequency Selective Fading Channels*, Internal Report, The NZi3 Wireless Research Centre, Christchurch, Jan. 2010.

[14] C. Horn, *Simulcast Receiver*, Internal Report, The NZi3 Wireless Research Centre, Christchurch, Jan. 2010.

[15] S. Haykin, *Digital Communications*, John Wiley & Sons, 1988.

[16] M. V. Eyuboglu and S. U. H. Qureshi, *Reduced-State Sequence Estimation with Set Partitioning and Decision Feedback*, IEEE Trans. Commun., vol. 36, pp. 1320, Jan. 1988.

[17] A. Duel-Hallen and C. Heegard, *Delayed Decision-Feedback Sequence Estimation*, IEEE Trans. Commun., vol. 37, pp. 428-436, May 1989.

[18] P. R. Chevillat and E. Eleftheriou, *Decoding of Trellis-Encoded Signals in The Presence of Intersymbol Interference and Noise*, IEEE Trans. Commun., vol. 37, pp. 669676, July 1989.

[19] T. Hashimoto, *A List-type Reduced-Constraint Generalization of the Viterbi Algorithm*, IEEE Trans. Inform. Theory, vol. IT-33, pp. 866-876, Nov. 1987.

[20] N. Sesliadri and C-E.W. Sundberg, *Generalized Viterbi Algorithms for Error Detection with Convolutional Codes*, Proc. Globecom, Dallas, Texas, pp. 1534-1538, Nov. 1989.

[21] F.K. Soong and E.F. Huang, *A Tree-Trellis Based Fast Search Algorithm for Finding the N Best Sentence Hypotheses in Continuous Speech Recognition*, Proc. IEEE Int. Conf. ASSP, pp. 705-708, 1991.

[22] N. Seshadri and C-E.W. Sundberg, *List Viterbi Decoding Algorithms with Applications*, IEEE Trans. Commun., rev. Mar. 1992.

[23] V. Ziablov, V. Potapov and V. Sidorenko, *Decoding of Convolutional Code into List*, Proc. Int. Workshop Voneshta Voda, Bulgaria, pp. 161-164, June 1992.

[24] P. Hoeher, *Advances in Soft-Output Decoding*, in Proc. Globecom, vol. 2, Houston, TX, pp. 793 797, Nov. 1993.

[25] C. Nill and C.-E. W. Sundberg, *List and Soft Symbol Output Viterbi Algorithms: Extensions and Comparisons*, IEEE Trans. Commun., vol.43, pp. 277287, Feb.-Apr. 1995.

[26] J. S. Sadowsky, *A Maximum Likelihood Decoding Algorithm for Turbo Codes*, in Proc. Globecom, vol. 2, pp. 929933, Dallas, TX, Nov. 1997.

[27] K. R. Narayanan and G. L. Stuber, *List Decoding of Turbo Codes*, IEEE Trans. Commun., vol. 46, pp. 754762, Jun. 1998.

[28] J. Hagenauer and P. Hoeher, *A Viterbi Algorithm with Soft-Decision Outputs and its Applications*, Globecom, Dallas, Texas, pp. 47.1.1-47.1.7, Nov. 1989.

[29] J. Turner, *Reduced Complexity Decoding of Space Time Trellis Codes in the Frequency Selective Channel*, Master Thesis, Canterbury University, May, 2005.

[30] L. Fang, J. M. Kuang and L. V. Xin, *Multi-Order Soft Output Viterbi Algorithm (SOVA) Decoder Fix Point Performance Simulation*, in Proc. ICII, Beijing, vol.2, pp. 151-155, 2001.

[31] P. Hoeher, *TCM on Frequency-Selective Fading Channels: A Comparison of Soft-Output Probabilistic Equalizers*, in Proc. Globecom, San Diego, CA, pp. 376381 Dec. 1990.

[32] J. Tan and G. L. Stuber, *A MAP equivalent SOVA for Non-binary Turbo Codes*, Proc. ICC, vol.2, pp. 602- 606, 2000.

[33] L. Cong, W. Xiaofu and Y. Xiaoxin, *On SOVA for nonbinary codes*, IEEE Commun. Lett., vol. 3, no. 12, pp. 335337, Dec. 1999.

[34] J. Liu and G. Tu, *Iterative Decoding of Non-binary Turbo Codes Using Symbol Based SOVA Algorithm*, Proc. ICC, vol.2, pp. 689 - 693, 2006.

[35] F. Labeau, *Low-Complexity Nonbinary SOVA for Sectionalized Trellises*, in Proc. IEEE WCNC., vol.4, pp. 2274-2279, 2004.

[36] J. Turner and D. Taylor, *Reduced Complexity Decoding of Space Time Trellis Codes in the Frequency Selective Channel*, IEEE Trans. Commun., vol. 57, no. 3, pp. 635-640, Mar. 2009.

[37] L. R. Bahl, J. Cocke, F. Jelinek and J. Raviv, *Optimal decoding of linear codes for minimizing symbol error rate*, IEEE Trans. Inform. Theory, pp. 284-287, Mar. 1974.

[38] S. H. Muller, W. H. Gerstacker and J. B. Huber, *Reduced-State Soft Output Trellis-Equalization Incorporating Soft Feedback*, in Proc. Globecom, London, U.K., pp. 95100, Dec. 1996.

[39] G.-K. Lee, S. B. Gelfand and M. P. Fitz, *Bayesian Decision Feedback Techniques for Deconvolution*, in Proc. Globecom, San Francisco, CA, pp. 248252, Dec. 1994.

[40] M. P. Fitz and S. B. Gelfand, *Reduced Complexity Symbol-by-Symbol Soft Output Algorithms*, in Proc. Globecom, London, U.K., pp. 2125, Dec. 1996.

[41] R. Balasubramanian, M. P. Fitz and J. K. Krogmeier, *Optimal and Suboptimal Symbol-by-Symbol Demodulation of Continuous Phase Modulated Signals*, IEEE Trans. Commun., vol. 46, pp. 16621668, Dec. 1998.

[42] X. Chen and K. M. Chugg, *Reduced-State Soft-Input/Soft-Output Algorithms for Complexity Reduction in Iterative and Noniterative Data Detection*, in Proc. ICC, New Orleans, LA, pp. 610, June 2000.

[43] G. Colavolpe, G. Ferrari and R. Raheli, *Reduced-State BCJR-Type Algorithms*, in Proc. IEEE Int. Conf. Commun., New Orleans, LA, pp. 460464, June 2000.

[44] D. G. Brennan, *Linear Diversity Combining Techniques*, Proc. IEEE, vol. 91, pp. 331-356, 2003.

[45] K. Young-Chai, L. Tao and J. Gibong, *Effect of Noise Variance in The Channel Estimation on Dual-MRC over Rayleigh Fading Channels*, in Proc. VTC, vol. 4, pp. 2538-2542, 2003.

[46] L. B. M. Ning-Kong, *Average SNR of a Generalized Diversity Selection Combining Scheme*, IEEE Commun. Letters, vol. 3, Mar. 1999.

[47] M. R. Chaaban, Y. H. Chung and A. M. D. Turkmani, *Diversity Reception at The Base Station of a GSM/DCS1800 System*, in Proc. IEEE VTC, vol. 1, pp. 709-713, 1994.

[48] A. F. Molisch, H. Novak, J. F. Fuhl and E. Bonek, *Reduction of The Error Floor of MSK by Selection Diversity*, IEEE Trans. Vehi. Tech., vol. 47, pp. 1281-1291, 1998.

[49] I. Ghareeb and S. Abu-Surra, *Differential Detection of GMSK Signals with Postdetection MRC over Correlated and Unbalanced Nakagamy Fading Channels*, IEEE Commun. Proc., vol. 152, pp. 221-228, 2005.

[50] I. Korn and J. P. Fonseka, *GMSK with Limiter-Discriminator Detector in Nakagami Fading Channel with and without Selection Combining*, IEEE Trans. Commun., vol. 51, pp. 1271-1273, 2003.

[51] W. Refai and S. C. Gupta, *Space Diversity of CPM over Fading Channels with Interfering Signals*, in Proc ICC, vol. 3, pp. 1641-1645, 1988.

[52] F. Adachi and J. D. Parsons, *Error Rate Performance of Digital FM Mobile Radio with Postdetection Diversity*, IEEE Trans. Commun., vol. 37, pp. 200-210, 1989.

[53] I. Korn, *M-ary CPFSK-DPD with L-diversity Maximum Ratio Combining in Rician Fast-Fading Channels*, IEEE Trans. Veh. Tech., vol. 45, pp. 613-621, 1996.

[54] L. Zhou, P. A. Martin, D.P. Taylor and C. Horn, *MLSE Diversity Receiver for Partial Response CPM*, IEEE ICCS, Vol. 17, pp. 501-505, Nov. 2010.

[55] J. M. Wozencraft and I. M. Jacobs, *Principles of Communication Engineering*, New York: Wiley, 1965.

[56] S. Haykin, *Communication Systems*, 4th ed., John Wiley & Sons, 2001.

[57] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*, Cambridge University Press, 2005.

[58] T. S. Rappaport, *Wireless Communications Principles and Practice*, 2nd ed., New York: Prentice-Hall, 2002.

[59] J. Cheng and T. Berger, *Performance analysis for MRC and Postdetection EGC over Generalized Gamma Fading Channels*, in Proc. WCNC. vol. 1, pp. 120-125, 2003.

[60] J. G. Proakis, *Digital Communications*, 4th ed. New York: McGraw-Hill, 2001.

[61] M. Nakagami, *The m-distribution: A general of Internsity Distribution of Rapid Fading*, Pergammon Press, 1960.

[62] M. C. Jeruchim, P. Balaban and K. S. Shanmugan, *Simulation of Communication Systems*, 2nd ed., New York, Kluwer Academic/Plenum, 2000.

[63] W. C. Jakes, *Microwave Mobile Communications*, New York: John Wiley & Sons, 1974.

[64] R. H. Clarke, *A statistical theory of mobile radio reception*, Bell Sys. Tech. J, vol. 47, no. 6, pp. 9571000, July/Aug. 1986.

[65] C.-P. Liang, J. Jong, W. E. Stark and J. East, *Nonlinear amplifier effects in communications systems*, IEEE Trans. Microwave Theory Tech., vol. 47, no. 8, pp. 14611466, Aug. 1999.

[66] T. Kailath, *A General Likelihood-ratio Formula for Random Signals in Gaussian Noise*, IEEE Trans., Inform. Theory, vol. IT-15, pp. 350-361, May 1969.

[67] G. Reise, G. Matz and J. Maurer, *MIMO Receivers Using Soft Information*, Mater Thesis, Institut fr Nachrichtentechnik und Hochfrequenztechnik, Nov. 2007

[68] J. K. Cavers, *Mobile Channel Characteristics*, 2nd Ed,. Columbia, Canada: Shady Island Press, 2003.

[69] P. Balaban and J. Salz, *Dual Diversity Combining and Equalization in Digital Cellular Mobile Radio*, IEEE Trans. Veh. Tech., vol. 40, May 1991.

[70] P. Reid, *Diversity for an Emerging Digital Radio Standard*, A Third Professional Year Project Report, University of Canterbury, Sep. 2002.

[71] L. Zhou, P.A. Martin, D.P. Taylor and C. Horn, *MLSE Diversity Receiver for Partial Response CPM*, in Proc. ICCS, Singapore, 17-20 Nov. 2010.

[72] M. J. Gertsman and J. H. Lodge, *Symbol-by-symbol MAP Demodulation of CPM and PSK signals on Rayleigh Flat-Fading Channels*, IEEE Trans. Commun., vol. 45, pp. 788-799, 1997.

[73] J. Lodge and M. Moher, *Maximum Likelihood Sequence Estimation of CPM Signals Transmitted Over Rayleigh Flat-Fading Channels*, IEEE Trans. Commun., vol.38, no.6, pp. 787794, Jun. 1990.

[74] J.K. Cavers, *Maximum Likelihood Antenna Combining in Fading Dispersive Channels*, Private Correspondence, 1994.

[75] X. Yu and S. Pasupathy, *Innovations-based MLSE for Rayleigh Fading Channels*, IEEE Trans. Commun., vol. 43, no 234, pp. 1543-1544, Feb/Mar/Apr. 1997.

[76] B.D. Hart and S. Pasupathy, *Innovations-Based MAP Detection for Time-Varying Frequency-Selective Channels*, IEEE Trans. Commun., vol. 48, no. 9, pp. 1507-1519, Sep. 2000.

[77] W.S. Leon and D.P. Taylor, *An Adaptive Receiver for the Time and Frequency-Selective Fading Channel*, IEEE Trans. Commun., vol. 45, no. 12, pp. 1548-1555, Dec. 1997.

[78] E. C. Ifeachor and B. W. Jervis, *Digital Signal Processing: A Practical Approach*, 2nd ed., Addison Wesley, 2002.

[79] A. Ambardar, *Analog and Digital Signal Processing*, 2nd ed., Brooks/Cole, 1999.

[80] L. R. Rabiner, *A tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*, in Proc. VTC, vol. 77, pp. 2173 - 2179, Feb. 2000.

[81] R. Raheli, A. Polydoros and C. K. Tzou, *Per-Survivor Processing: A General Approach to MLSE in Uncertain Enviroments*, IEEE Trans. Commun., vol.43, no.2, pp. 354-364, Feb. 1995.

[82] R. Raheli, A. Polydoros and C. K. Tzou, *The principle of Per-Survivor Processing: A General Approach to Approximate and aAdaptive MLSE*, Proc. Globecom, pp. 1 6, 1991.

[83] M. J. Omidi, S. Pasupathy and P. G. Gulak, *Joint Data and Kalman Estimation for Rayleigh Fading Channels*, Wireless Pers. Commun., vol. 10, pg 319-339, 1999.

[84] N. Seshadri, *Joint Data and Channel Estimation Using Blind Trellis Search Techniques*, IEEE Trans. Commun., vol. COM-42, part I, pp. 1000-1011, Apr. 1994.

[85] A. Reichman and R. A. Scholtz, *Joint Phase Estimation and Data Decoding for TCM Systems*, Proc. First Int. Symp. Com. Theory and Applications, Scotland, U.K., Sept. 1991.

[86] A. N. DAndrea, U. Mengali and G. M. Vitetta, *Detection of Coded PSK Signals with Unknown Carrier Phase*, Proc. Fifth Int. Workshop on Digital Commun., Tirrenia, Italy, pp. 413-422, Sept. 1991

[87] A. N. D Andrea, U. Mengali and G. M. Vitetta, *Approximate ML Decoding of Coded PSK with No Explicit Carrier Phase Reference*, IEEE Trans. Commun., vol. COM-42, part I, pp. 1033-1039, Apr. 1994.

[88] M. V. Eyuboglu and S. U. H. Qureshi, *Reduced-State Sequence Estimation for Coded Modulation on Intersymbol Interference Channels*, IEEE Journal Selected Areas in Commun., vol. 7, pp. 989-995, Aug. 1989.

[89] K. R. Narayanan and G. L. Stuber, *Performance of Trellis-Coded CPM with Iterative Demodulation and Decoding*, IEEE Trans. Commun., vol. 49, no. 4, pp. 676-687, April 2001.

[90] X. Rui, Z. Dan-feng and X. Chun-li, *A Novel Approach to Improve the Iterative Detection Convegence of LDPC Coded CPM Modulated Signal*, Proc. WiCOM Conf., pp. 1-5, Sept. 2010.

[91] A. F. Molisch, J. Fuhl, and P. Proksch, *Bit Error Probability of MSK Modulation with Switched Diversity in a Mobile-Radio Channel with Two Independently-Fading Paths*, in Proc. PIMRC, vol. 3, pp. 1223, 1995.

[92] M. K. Simon and M. S. Alouini, *Digital Communication over Fading Channels: A Unified Approach to Performance Analysis*, NY: Wiley-Interscience, 2000.

[93] L. Zhou, *Low Complexity PSP-MLSE Receiver for H-CPM with Receive Diversity*, Master Thesis, Canterbury University, May, 2009.

[94] G. L. Stuber, *Principles of Mobile Communication*, Norwell, MA: Kluwer Academic Publishers, 1996.

[95] K. Young-Chai, L. Tao and J. Gibong, *Effect of Noise Variance in The Channel Estimation on Dual-MRC over Rayleigh Fading Channels*, in Proc. VTC, vol. 4, pp. 2538-2542, 2003.

[96] L. B. M. N. Kong, *Average SNR of a Generalized Diversity Selection Combining Scheme*, IEEE Commun. Letters, vol. 3, March 1999.

[97] Z. Sharif and A. Z. Shaameri, *Estimation of The Doppler Spread and Time Delay Spread for The Wireless Communication Channel*, in Proc. ICCAIE, pp. 438-442, Dec. 2010.

[98] K. Witrisal, Y. H. Kim and R. Prasad, *RMS Delay Spread Estimation Technique Using Non-Coherent Channel Measurements*, IEEE Commun. Letters, vol. 34, pp. 1918-1919, Oct. 1998.

[99] H. Arslan and T. Yucek *Delay Spread Estimation for Wireless Communication Systems*, in Proc. ISCC, vol. 1, pp. 282-287, July 2003.